

HCL DevOps Test Hub
11.0.1 Documentation
March 2024



Special notice

Before using this information and the product it supports, read the information in [Notices on page mxo](#).

Contents

Chapter 1. Release Notes.....	9
What's New.....	10
Deprecations and removals.....	38
Defect fixes.....	40
Known issues.....	52
Chapter 2. System Requirements.....	79
Hardware.....	80
Operating systems and Containers.....	80
Host prerequisites.....	82
Supported software.....	82
Chapter 3. Getting Started Guide	87
Overview of DevOps Test Hub.....	87
Supported versions of assets and resources.....	88
Suites and tests that are supported on DevOps Test Hub.....	89
Virtual services that are supported on DevOps Test Hub.....	91
Accessibility features of DevOps Test Hub.....	92
Chapter 4. Administrator Guide.....	93
Installation of DevOps Test Hub.....	93
Installation of DevOps Test Hub on Azure Kubernetes Service.....	94
Installation of DevOps Test Hub on RHEL.....	104
Installation of DevOps Test Hub on Ubuntu.....	107
Management of DevOps Test Hub features.....	111
DevOps Test Hub upgrade methods.....	114
DevOps Test Hub upgrade on Ubuntu.....	114
Upgrading DevOps Test Hub on Azure Kubernetes Service.....	115
Migration of user data.....	119
Migrating user data on Ubuntu from 10.1.2 or earlier.....	122
Migrating user data on Ubuntu from 10.1.3 or later.....	123
Backup and restoration of the DevOps Test Hub data.....	125
Backup and restoration of DevOps Test Hub data on Ubuntu.....	126
Backup and restoration of DevOps Test Hub data on Azure Kubernetes Service.....	130
Uninstallation of DevOps Test Hub.....	135
Uninstalling DevOps Test Hub from Ubuntu.....	135
Uninstalling DevOps Test Hub from Azure Kubernetes Service.....	136
Configuration of DevOps Test Hub.....	137
User administration.....	137
Managing third-party licenses.....	155
Certificate authority: Importing and extending lists.....	156
Copying third-party application Jars to Kubernetes.....	160
Changing the password seed.....	162
Server extensions.....	164
Team space administration.....	168
Team space overview.....	168
Creating a team space.....	171
Tasks in a team space.....	173

Tasks in projects in a team space.....	177
HCL license portal.....	178
Chapter 5. Test Author Guide.....	189
Edit branch overview.....	189
Data overview.....	190
Data navigator panel.....	190
Prerequisites for working with Datasets and Data definition.....	192
Management of classifications.....	193
Management of database connections.....	197
Management of database query.....	206
Working with Datasets on DevOps Test Hub.....	216
Creating a dataset.....	217
Creating a dataset by using a CSV file.....	219
Creating a dataset by using a Data definition.....	221
Creating a dataset by using a database query.....	224
Editing a dataset.....	226
Dataset encryption.....	233
Publishing changes in datasets.....	236
Viewing a dataset.....	237
Deleting a dataset.....	240
Viewing an encrypted dataset.....	241
Changing classification for an encrypted dataset.....	242
Managing an encrypted dataset.....	243
Data fabrication.....	246
Overview of data fabrication.....	246
Creating an empty Data definition.....	251
Creating a Data definition by importing fields from a CSV file.....	253
Creating a Data definition by importing fields from a dataset.....	255
Creating a Data definition by importing fields from a database query.....	258
Saving a Data definition by adding basic generators with their default settings.....	260
Saving a Data definition by adding basic generators with their modified settings.....	262
Modifying a Data definition.....	264
Publishing changes in an Edit branch.....	267
Viewing a Data definition and its details.....	269
Deleting Data definitions.....	273
Basic generators.....	273
Generation of test data.....	323
Management of custom generators.....	328
Management of custom categories.....	335
Chapter 6. Test Execution Specialist Guide.....	339
System modeling.....	339
Task flows for working with a system model.....	343
Adding a repository to a team space.....	345
Creating a system model.....	347
Creating components.....	349
Creating child components.....	350
Creating linkages between components.....	352

Deleting linkages between components.....	354
Modifying components.....	356
Deleting components.....	358
Publishing changes to the system model.....	359
Associating resources with components.....	359
Viewing resources that are associated with components.....	361
Removing resources that are associated with components.....	365
Publishing changes of associating or dissociating resources with components.....	367
Viewing the system model.....	369
Deleting a system model.....	369
Deleting a repository that is added to a team space.....	370
Prerequisites to running tests.....	371
Test run considerations for AFT Suites or Functional tests.....	371
Test run considerations for API Suites or API tests.....	373
Test run considerations for browser-based Web UI tests.....	376
Test run considerations for JMeter tests.....	376
Test run considerations for JUnit tests.....	378
Test run considerations for using Machine learning capabilities.....	379
Test run considerations for Postman tests.....	380
Test run considerations for schedules.....	382
Test run considerations for Selenium tests.....	383
Test run considerations for using mobile device clouds.....	384
Management of agents.....	386
Management of performance agents.....	386
Management of API agents.....	394
Management of Docker hosts.....	401
Test run considerations for running tests on remote Docker hosts.....	401
Setting up a remote Docker host.....	403
Setting up a secure remote Docker host.....	404
Copying third-party application Jars to a remote Docker host.....	406
Registering a remote Docker host.....	408
Viewing remote Docker hosts that are registered with DevOps Test Hub.....	410
Adding a remote Docker host.....	411
Editing configurations.....	412
Deleting a remote Docker.....	414
Unregistering a remote Docker host from DevOps Test Hub.....	416
Management of remote Kubernetes clusters.....	417
Considerations for using a remote Kubernetes cluster.....	417
Registering a remote Kubernetes cluster.....	420
Viewing remote Kubernetes clusters.....	423
Adding a remote Kubernetes cluster to a project.....	425
Editing configurations of a remote Kubernetes cluster.....	427
Removing a remote Kubernetes cluster from a project.....	429
Unregistering a remote Kubernetes cluster from a team space.....	430
Management of mobile device cloud credentials.....	432
Adding cloud credentials of mobile device clouds.....	435
Adding authorized users to a cloud credential.....	438

Viewing details of cloud credentials.....	439
Editing details of cloud credentials.....	442
Removing authorized users from a cloud credential.....	444
Deleting an existing cloud credential.....	445
Test run configurations.....	446
Configuring a Suite or test for a quick run.....	449
Configuring an AFT Suite run.....	453
Configuring an AFT Suite to run on a mobile device by using variables.....	466
Configuring an AFT Suite to run on a mobile device by using a cloud credential.....	478
Configuring an API Suite or an API test run.....	486
Configuring a run of a Compound Test that contains HTML tests.....	498
Configuring a run of a Compound Test that contains mobile tests to run by using variables.....	509
Configuring a run of a Compound Test that contains mobile tests to run by using a cloud credential.....	523
Configuring a run of a Compound Test that contains Web UI tests.....	533
Configuring a run of a Compound Test that contains Web UI tests to run by using variables.....	545
Configuring a run of a Compound Test that contains Web UI tests to run by using a cloud credential.....	559
Configuring a run of a Compound Test that contains performance tests.....	569
Configuring a run of a mobile test to run by using variables.....	581
Configuring a run of a mobile test to run on a mobile device by using a cloud credential.....	595
Configuring a run of a Performance test.....	605
Configuring a run of a Web UI test.....	617
Configuring a run of a Web UI test to run by using variables.....	629
Configuring a run of a Web UI test to run by using a cloud credential.....	643
Configuring a run of a Windows test.....	653
Configuring a run of a JMeter test.....	665
Configuring a run of a JUnit test.....	674
Configuring a run of a Postman test.....	683
Configuring a run of a Rate Schedule or VU Schedule.....	692
Configuring a run of a Selenium test.....	704
Configuration of test runs in a GitHub Actions Workflow.....	713
Prerequisites to run tests in a GitHub Actions Workflow.....	713
Configuring a test run in a GitHub Actions Workflow.....	714
Configuration of a scan of an application code.....	716
Considerations to scan an application code.....	717
Configuring a run to scan the application code.....	719
Management of running tests.....	727
Viewing the state of test assets.....	727
Viewing the progress of running test assets.....	731
Checking logs.....	734
Resetting the configuration settings for a test run.....	735
Stopping test runs.....	736
Canceling scheduled test runs.....	739
Management of virtualized services.....	741
Prerequisites for running HTTP virtual services.....	742
Viewing intercepts that are registered with a team space on DevOps Test Hub.....	746
Viewing virtual service resources.....	747
Configuring a run of a virtual service.....	750

Running HTTP virtual services in Kubernetes cluster without using proxies.....	765
Viewing running instances of virtual services.....	770
Viewing configurations of running instances of virtual services.....	775
Modifying configurations of running instances of virtual services.....	780
Viewing routing rules of the virtual services.....	783
Viewing usage statistics of virtual services.....	784
Stopping virtual services.....	787
Test results.....	788
Test results and reports.....	789
Test results data in Grafana.....	801
Viewing Surefire format reports.....	807
Test logs.....	808
Test log queries	823
Machine learning capabilities in DevOps Test Hub.....	826
Viewing Machine learning insights.....	829
Interpretation of insights.....	832
Viewing Machine learning analyzers.....	834
Interpretation of parameter analysis.....	836
Viewing reactions marked by members to Machine learning insights.....	839
Modifying Machine learning analyzer settings.....	841
Deleting Machine learning insights.....	843
Rerunning Machine learning (ML) analysis.....	844
Resource monitoring.....	845
Resource monitoring capabilities.....	846
Resource monitoring agents.....	864
Managing sources.....	876
Collecting and viewing resource monitoring usage data.....	883
Configuration of a change management system.....	891
Configuration of Atlassian Jira as a change management system.....	892
Configuration of Azure DevOps as a change management system.....	904
Configuration of Rational® ClearQuest® or HCL® Compass as a change management system.....	914
Integrations with other applications.....	924
Integration plugin compatibility matrix.....	924
Integration with Azure DevOps.....	926
Integration with HCL DevOps Deploy.....	936
Integration with IBM® DevOps Deploy.....	952
Integration with IBM® Engineering Test Management.....	967
Integration with Jenkins.....	973
Integration with Xray test management for Jira.....	988
Managing access to DevOps Test Hub.....	994
Working with DevOps Test Hub REST services.....	996
Chapter 7. Test Manager Guide.....	997
Managing team spaces.....	997
Viewing team spaces.....	997
Adding members to a team space.....	998
Modifying the configuration of a team space	1001
Viewing the configuration of a team space	1003

Becoming a team space member.....	1005
Managing members and their roles in a team space.....	1006
Deleting a team space.....	1009
Managing DevOps Test Hub projects.....	1010
Working with projects in a team space.....	1010
Repository considerations for a server project.....	1011
Adding a project.....	1013
Adding repositories to a server project.....	1014
Secrets configuration	1017
Adding users to a project.....	1017
Viewing active projects.....	1018
Changing the visibility configuration of a project.....	1019
Becoming a project member.....	1020
Managing access to the server projects.....	1021
Archiving or unarchiving server projects.....	1023
Deleting server projects.....	1024
Project overview.....	1025
Managing repositories.....	1026
Adding repositories to a server project.....	1027
Updating the authentication credentials of the repository.....	1030
Deleting a repository.....	1031
Refreshing repositories manually.....	1031
Selecting the global branch in a project.....	1033
Creating webhooks.....	1034
Protecting API test assets by using secrets.....	1035
Creating a secret in a classification.....	1035
Granting access to members or member roles.....	1037
Managing secrets.....	1038
Managing notifications.....	1040
Managing in-app notifications.....	1041
Managing email notifications.....	1047
Configuration of a webhook.....	1051
Chapter 8. Troubleshooting.....	1077
Troubleshooting issues.....	1077
Security Considerations.....	mlxxxviii
Notices.....	mxc
Index.....	1094

Chapter 1. Release notes for DevOps Test Hub 2024.03 (11.0.1)

This document includes information about the new features introduced, features deprecated or removed, defects fixed, and known issues identified in this version of HCL DevOps Test Hub (Test Hub). You can also find the installation and upgrade instructions along with the contact information of HCL Customer Support.

Product description

Test Hub is a server that includes capabilities such as project and role-based security, Docker-based distribution and installation, and running of test cases. For more information about the server, see [Overview of DevOps Test Hub on page 87](#).

What's new

You can find information about the features introduced, enhancements, or other changes in Test Hub. See [What's New on page 10](#).

Deprecated features or capabilities

To find information about the features or capabilities that are deprecated in this version and are planned for removal in a future release, see [Deprecations and removals on page 38](#).

Removed features or capabilities

To find information about the features or capabilities that are removed in this version, see [Deprecations and removals on page 38](#).

Product download and installation

If you have purchased the licenses to use the product, you can download the product software packages from [HCL® License & Delivery portal](#).

For instructions about installing the product software, see [Installation of DevOps Test Hub on page 93](#).

Product upgrade

For instructions about upgrading the product software, see [DevOps Test Hub upgrade methods on page 114](#).

Defect fixes and known issues

For the list of defects fixed in this version that include defects submitted by customers and defects found during testing, see [Defect fixes on page 40](#).

For the list of known issues that are identified in this version and the previously published known issues that are still applicable, see [Known issues on page 52](#).

Contacting HCL support

- For technical assistance, contact [HCL Customer Support](#).
- Before you contact HCL support, you must gather the background information that you might need to describe your problem. When you describe a problem to an HCL Support specialist, be as specific as possible and include all relevant background information so that the specialist can help you solve the problem efficiently.

To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, or messages that are related to the problem?
- Can you reproduce the problem? If so, what steps do you take to reproduce it?
- Is there a workaround for the problem? If so, be prepared to describe the workaround.

What's New

You can find information about the features introduced, enhancements, or other changes made in HCL DevOps Test Hub (Test Hub). You can also find the features that were introduced in earlier versions of Test Hub. DevOps Test Hub is the new name of HCL OneTest™ Server from 11.0.0 onwards.

What's New in DevOps Test Hub 2024.03 (11.0.1)

The following section lists the features, enhancements, or other changes made in Test Hub 2024.03 (11.0.1).

Feature title	Description
Adding an external database connection and importing data to create test assets	You can now add an external database connection to a project in Test Hub, and then create a database query to import data from the database. You can also create test assets such as datasets and Data definitions by using the database query.
Enhanced the UI with the new Security page	<p>You can now use the Security page to create and manage API secrets and dataset classifications. You can also add the credentials of external databases to data classifications.</p> <p>You can now provide Read access to the classifications at the role level, while Write access is restricted. The Access Control List UI is enhanced so that you grant both Read and Write access to members with classification privileges. See Protecting API test assets by using secrets on page 1035, Managing an encrypted dataset on page 243.</p>
Installing and configuring the DevOps Test Hub Results Data plugin by using a proxy connection	You can now install and configure the DevOps Test Hub Results Data plugin by using a proxy connection in Grafana instead of using additional helm parameters during the installation of Test Hub. See Installing and configuring the DevOps Test Hub Results Data plugin on page 802 .

Feature title	Description
Viewing resource monitoring usage data on Grafana	You can now view the resource monitoring usage data by creating queries on Grafana, and retrieve this data that is collected for a certain period during a test run in Test Hub. See Viewing resource monitoring usage data on Grafana on page 887 .

What's New in earlier versions

You can find information about the features introduced in earlier versions of Test Hub in the following sections.

What's New in 11.0.0

The following section lists the features, enhancements, or other changes made in Test Hub 11.0.0.

Feature title	Description
Renaming of OneTest products	<p>The following OneTest products have new names starting from 11.0.0:</p> <ul style="list-style-type: none"> • HCL OneTest™ Server is now HCL DevOps Test Hub. • HCL OneTest™ UI is now HCL DevOps Test UI. • HCL OneTest™ API is now HCL DevOps Test Integrations and APIs. • HCL OneTest™ Performance is now HCL DevOps Test Performance. • HCL® Quality Server is now HCL DevOps Test Virtualization Control Panel. • HCL® OneTest™ Embedded is now HCL DevOps Test Embedded. • HCL OneTest™ Virtualization is now HCL DevOps Test Virtualization. • HCL OneTest™ Studio is now HCL DevOps Test Workbench.
Enhancements to the data fabrication feature	<p>The data fabrication feature is enhanced and you can now perform the following tasks:</p> <ul style="list-style-type: none"> • Create an empty Data definition, or import the header row fields from a CSV file or a dataset. You can then add the basic generators or modify the generators to suit your test data fabrication requirements. • Create folders in your project repository and select them as locations to save your data definitions. • Use the Save All option to save all files that are opened for editing or have unsaved changes. • Use the right-click options to close any or all files that are opened for editing or have unsaved changes. • Use the distribution parameter, which is enhanced to automatically allocate and rebalance the distribution among the existing and new configurations that are added subsequently. • Preview the sample test data for each configuration of a generator.

Feature title	Description
Changes to the Licensing environment variables	With the renaming of HCL OneTest Products to HCL DevOps Test Products in 11.0.0, the HCL Licensing environment variables that contained <i>ONETEST</i> are changed to be used without <i>ONETEST</i> . For example, the <i>HCL_ONETEST_LICENSING_URL</i> environment variable now must be called <i>HCL_LICENSING_URL</i> . Note that the old environment variable names are still supported.
Support to view configuration details of a Virtual Service instance	You can now view the configuration details of a Virtual Service that you configured during a run from the Instances page. The details such as the datasets selected for overrides, variables, JVM arguments, environment variables, and additional parameters are displayed, if they are configured.
Enhancements to editing datasets	The editing capabilities of datasets are further enhanced in Test Hub 11.0.0. You can now perform different actions to edit the dataset from the UI by using the Editing panel, the context menu options, and the insert options available in the Dataset panel.
New options to edit and save Data definitions	You can now use the right-click options for Data definitions from the Data Navigator to open the Data definition for editing or saving it as a different Data definition.

What's New in 10.5.4

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.5.4.

Feature title	Description
Generation of test data enhanced through redesign	<p>The test data generation that was previously possible by using HCL® OneTest™ Data is enhanced through the redesign of the user interface and the supported architecture and is now available in a new avatar.</p> <p>You can use the in-built data generators or create customized data generators to suit your test data requirements. You can preview the test data in the UI before you generate the test data. You can also download the generated test data.</p> <p>You can view the history of the test data generated, and you can regenerate the test data with or without altering any of the parameters used.</p> <p>See Data fabrication on page 246.</p>
Support for Machine learning insights on performance tests	HCL OneTest™ Server is enhanced to provide Machine learning (ML) insights on performance tests that match certain criteria when you run tests from projects in a team

Feature title	Description
	<p>space. HCL OneTest™ Server uses ML algorithms to analyze the following types of parameters in performance tests:</p> <ul style="list-style-type: none"> • Response Time Lock-Step Pattern • Response Time Standard Deviation Pattern • Throughput Drop Pattern <p>The findings from the analysis are presented as a recommendation for each of the parameters on the Infrastructure > Analyzers page in the team space.</p> <p>You can improve ML recommendations by altering the default Threshold value that specifies the level of confidence required in the insights. You can also improve the accuracy of the recommendations by running multiple tests with differing loads repeatedly.</p> <p>See Machine learning capabilities in DevOps Test Hub on page 826.</p>
<p>Enhanced the dataset editor and managing capabilities</p>	<p>The CSV Editor that is used to edit datasets is enhanced and now opens within the HCL OneTest™ Server UI, which previously opened in a separate browser tab.</p> <p>Further, the pages and the methods to create, open, or view datasets are changed in their presentation in the UI.</p> <p>See Working with Datasets on DevOps Test Hub on page 216.</p>
<p>Enhanced dataset classification</p>	<p>You can now set the dataset classification (encrypted password) for a dataset irrespective of its location in branches in a repository. The dataset classification that you set is applied to all branches and you need not set a password for each branch that contains the dataset. If you want to use different passwords, then you can create a different dataset by cloning the dataset that you want.</p> <p>See Managing an encrypted dataset on page 243.</p>
<p>Support for in-built Chrome and Edge browsers</p>	<p>HCL OneTest™ Server now provides support for running browser-based tests by using the in-built Chrome and Edge browsers after you accept the license terms for usage. If you do not accept the license terms at the time of signing in or later, tests that use these browsers fail to run successfully.</p> <p>See Managing third-party licenses on page 155.</p>

Feature title	Description
Support to run AFT Suite on a mobile device by using variables	<p>You can now run an AFT Suite, which contains mobile or Web UI tests on a device, emulator, or simulator that is connected to an agent of a device cloud. You must enter the details of the agent or device cloud as variables while you configure a run of the AFT Suite.</p> <p>See Configuring an AFT Suite to run on a mobile device by using variables on page 466.</p>
Support to override the project in a mobile device cloud	<p>You can now select or create a project to override the project that is configured for a cloud credential when you want to run a mobile test, Web UI test, Compound test, or an AFT Suite. The tests run on mobile devices that are connected to a mobile device cloud.</p> <p>See Configuring an AFT Suite to run on a mobile device by using a cloud credential on page 478, Configuring a run of a Compound Test that contains mobile tests to run by using a cloud credential on page 523, or Configuring a run of a Compound Test that contains Web UI tests to run by using a cloud credential on page 559.</p>
Support to add reports and logs to Jira issues	<p>After you configure Jira as a change management system on HCL OneTest™ Server, you can now add the test reports or test log to the Jira issue that you create for a test result. You can then view the attached reports or log from your project in Jira.</p> <p>See Creating Jira issues for the test results on page 895.</p>
Support to view test log statistics in Grafana	<p>You can now view the overall statistics and time-series statistics in Grafana by using queries for the test logs that are generated in HCL OneTest™ Server.</p> <p>See Viewing test results data in Grafana on page 804.</p>
Enhanced the display of events in Web UI or Functional test logs	<p>The display of the events in the Web UI or Functional test log is enhanced. Previously, the events were displayed under the UIScript node and placed under the Test node, which did not represent the correct hierarchy.</p> <p>Now, the events are placed under the Test node and the UIScript node is no longer displayed, thus displaying the correct hierarchy of the events in the test log.</p>
Requirement of Java 17 for Resource Monitoring agents	<p>The Resource Monitoring Java Agent now requires Java 17 JRE or JDK to be installed on the computer that is used to collect the Resource Monitoring data from HCL OneTest™ Server.</p>

What's New in 10.5.3

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.5.3.

Feature title	Description
Support for Revenera 2022.12	With Revenera 2018.02 (LLS 1.0) reaching EOS, you can now use Revenera 2022.12 (LLS 3.0) as a local license server to host licenses for HCL OneTest™ Server.
Support to run mobile tests on device clouds	<p>You can now configure a connection from HCL OneTest™ Server to a mobile device cloud by using the credentials of your cloud subscription account. You can then add team space members as authorized users of the cloud credential. The authorized users can run mobile tests on the mobile devices that are connected to a mobile device cloud.</p> <p>See Management of mobile device cloud credentials on page 432.</p>
Enhancement to the webhooks feature	<p>The webhooks feature is enhanced to enable you to configure the behavior of notifications that were not sent because the webhook was disabled. You can decide to discard or send the queued notifications when you re-enable the webhook. Webhook templates defined at the team space are now also displayed at the project level.</p> <p>See Configuration of a webhook on page 1051.</p>
Support to download test reports	<p>You can now download the reports and logs of the tests from the Results page. Reports can be saved to a folder on your computer as a compressed file or PDF, which depends on the format of the generated report.</p> <p>See Test results and reports on page 789.</p>
Support to override input tags of virtual services	<p>You can now specify values for stub input tags by using advanced configuration parameters when you run a virtual service.</p> <p>See Additional configuration parameters for virtual services on page 759.</p>
Support to add an Azure repository	<p>You can now add an Azure repository that contains test resources to your project in HCL OneTest™ Server, and then run the resources from the server.</p> <p>See Repository considerations for a server project on page 1011.</p>

Feature title	Description
Removed support of Prometheus for virtual services	Virtual service usage in HCL OneTest™ Server no longer utilizes Prometheus. If you upgrade from a previous version of HCL OneTest™ Server to 10.5.3, any virtual service usage that was captured before 10.5.3 will not be visible in the team space or project virtualization usage pages.


What's New in 10.5.2

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.5.2.

Feature title	Description
Running Selenium tests on HCL OneTest™ Server	<p>You can now run Selenium tests that are within a Maven project from HCL OneTest™ Server.</p> <p>You must commit the Maven project as a <code>pom.xml</code> to a Git repository. After you add the Git repository to your project on HCL OneTest™ Server, you can configure a run of the Selenium test from the Execution page.</p> <p>See Configuring a run of a Selenium test on page 704.</p>
Running virtual services on API agents from HCL OneTest™ Server	<p>You can now configure an HCL OneTest™ API Agent to register with a team space in HCL OneTest™ Server by using the server hostname, team space name, and your offline token. When you start the virtual services, you can select the HCL OneTest™ API agent as the location and not select the Kubernetes cluster.</p> <p>See Management of API agents on page 394.</p>
Support for IBM® Rational® ClearQuest® as a change management system	<p>You can now create Rational® ClearQuest® work items for the tests that ran on HCL OneTest™ Server, after you configure Rational® ClearQuest® as a change management system on HCL OneTest™ Server.</p>
Configuring external webhooks on HCL OneTest™ Server	<p>You can configure the URL of third-party applications such as Teams or Slack, as a webhook on HCL OneTest™ Server when you want to receive push notifications about events that occur on the server.</p> <p>You can create templates and select specific events for which you want to receive notification messages in the defined template for those events.</p> <p>See Configuration of a webhook on page 1051.</p>

What's New in 10.5.1

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.5.1.

Feature title	Description
Support to export test results to a Jira Xray project	<p>When you use Atlassian Jira as a change management system on HCL OneTest™ Server and you use Xray as the test management application in Jira, you can now integrate Xray with HCL OneTest™ Server. With this integration, you can export the test results of the tests that run on HCL OneTest™ Server to an issue in your Jira Xray project. You can then view and use the test results in your project in Jira.</p> <p>See Integration with Xray test management for Jira on page 988.</p>
Copying of the generated keys individually	<p>When you configured Atlassian Jira as a change management system on HCL OneTest™ Server, you were able to copy the public and consumer keys in a single action. However, it was difficult to differentiate the keys when you pasted them in the individual fields on the Jira server configuration page. You can now copy the public and consumer keys separately and use them to configure the Jira server.</p>
Support to create defects on tests run by other project members	<p>Previously, you were able to create defects only on tests that you ran on HCL OneTest™ Server. You can now create defects in the change management system configured in your project on tests that are run by other project members too.</p>
Enhanced publishing of association changes in the system model	<p>Previously, you were able to publish the changes made to the component associations with test resources in the system model from the Author > Changes page. You can now review and publish the association or dissociation changes of test resources with components in a system model from the System Model page. You can use the  icon on the System Model page to publish the association changes.</p>

What's New in 10.5.0

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.5.0.

Feature title	Description
Support to install Jaeger as part of HCL OneTest™ Server is removed	<p>The support to install Jaeger during the installation of HCL OneTest™ Server is removed.</p>

Feature title	Description
Support for HCL® Compass as a change management system	You can now create HCL® Compass records for the tests that you run on HCL OneTest™ Server, after you configure the HCL® Compass server as a change management system on HCL OneTest™ Server.
Running tests on remote Kubernetes clusters	<p>You can now run the supported test resources on remote Kubernetes clusters that you register with a team space on HCL OneTest™ Server and add the Kubernetes cluster to your project in the team space.</p> <p>You can select the remote cluster as a location to run the tests when you configure a run.</p> <p>You can edit the configurations of a cluster, remove a cluster from a project, or unregister the cluster from a team space.</p> <p>See Management of remote Kubernetes clusters on page 417.</p>
Support to run tests or resources as a GitHub Action	<p>You can now run tests or resources that are contained in your project on HCL OneTest™ Server as a GitHub Action from a Git repository.</p> <p>See Configuration of test runs in a GitHub Actions Workflow on page 713.</p>
Creating an <i>Edit</i> branch to contain resources before committing to the repository branch	When you want to create or edit datasets, test assets, virtual service resources, or resources that are associated with components in a system model, you must now create an <i>Edit</i> branch, which is a temporary branch that will contain these until the time you commit or publish these to a branch in the project repository or the team space repository.
Support to schedule recurring test runs	You can now schedule recurring test runs on HCL OneTest™ Server.
Panning icon removed from the System Model page	The Pan icon that existed previously to pan the view in the System Model page is now removed. You can now pan the view by clicking the area and moving the area in the direction you want with the left-mouse button pressed.
Enhancements to the HCL OneTest™ Studio extension	<p>When you run tests from Azure DevOps Pipeline by using the HCL OneTest™ Studio extension, you can now store the JUnit report that is generated for the test in the XML format.</p> <p>See Running DevOps Test Hub tests in an Azure DevOps Pipeline on page 929.</p>

Feature title	Description
Enhancements to the Jenkin plugin	You can now view the test results by using the Test Results Analyzer feature in Jenkins if you have installed the Test Results Analyzer plugin in Jenkins. See Running tests from Jenkins on page 987 .

What's New in 10.2.3

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.2.3.

Feature title	Description
Changes to accessing the software from Harbor	You can now access the HCL OneTest™ Server Helm charts from the Harbor repository without providing Okta credentials.
Support to view or hide license details	As a system administrator or team space owner, you can now view or hide the details of the License Server on the License Configuration page.
Configuring Azure DevOps as a change management system	Previously, you might have added Atlassian Jira as a change management system on HCL OneTest™ Server, if you used Jira to create and monitor issues. You can now add Azure DevOps as a change management system on HCL OneTest™ Server, if you use Azure DevOps to create and monitor defects or bugs, issues, or other work items. After you configure Azure DevOps on HCL OneTest™ Server, you can create work items for the test runs from HCL OneTest™ Server and view the created work items on Azure DevOps by clicking the links within HCL OneTest™ Server.
Viewing the location where virtual services ran	On the Progress and Results pages, you can now view the location where the virtual service instance ran based on the location that you selected when you started a virtual service instance. The location displayed is the name of the remote Kubernetes cluster or the <i>Default Cluster</i> if the virtual service ran on HCL OneTest™ Server.
Enhancement to publishing of datasets from HCL OneTest™ Server	HCL OneTest™ Server now provides the capability to publish a dataset subsequent to publications from other users when the changes made to the datasets do not conflict with the changes made by the other users.
Viewing or editing load test profile details for Schedules	You can now view the load test profile details for the Schedules that you added in HCL OneTest™ Performance in the Description column on the Author > Tests page. You can also edit the load test profile details if you do not want to run the Schedules with the configured profile.

Feature title	Description
Adding notes to a test result	You can now add notes to a test result and group these notes into a discussion in the Details panel on the Results page. You can also perform a search for specific content in the notes.
Enhanced the option to export queries from HCL OneTest™ Server	Previously, you were able to export the queries that you created from HCL OneTest™ Server as a curl command. You can now use the Export option to export the queries to Shell, Curl, or PowerShell scripts. You can now select the script, the output format, and the number of results that are returned by the query.
Enhanced the viewing of test logs	<p>Previously, the test log displayed all the activities that were captured for a test run and viewing the essential information was tedious. You can now select a view that filters the activities that you might be interested in as a tester. You can however, view all the activities that are captured for a test run by switching off the filter.</p> <p>The Test Log page now displays the view correctly when you return to the page after viewing another page on the server UI. You can also use an option on the UI to view the test log list.</p>
Viewing attachments in the test log	You can now view the attachments such as images, documents, PDFs, scripts, json or xml files, that are captured by the test log. You can view the attachments that are supported for display in a web browser. You can also copy or download the attachments to your computer.
Viewing the test run duration and time from test logs	After you run a test, you can view the calendar time and the duration for each of the test steps from the test log.
Enhancements to integration plugins	<p>While you configure test runs from the integrated product, you can now use additional parameters such as: Datasets, Labels, Secret Collection, Start Date, and Variables in the following integration plugins:</p> <ul style="list-style-type: none"> • Azure DevOps • HCL® Launch • Jenkins
Support to run virtual services by using server plugins	Previously, you were able to run test assets that were in your project on HCL OneTest™ Server from the integrated applications by using the server plugins. You can now run or start instances of virtual service resources that are in your project on HCL

Feature title	Description
	OneTest™ Server from integrated applications such as Azure DevOps, Jenkins, and HCL® Launch.
Support to run additional test assets from Jenkins	<p>You can now run the following types of test assets from the Jenkins server by using the HCL OneTest™ Server Jenkins plugin:</p> <ul style="list-style-type: none"> • API tests • Functional tests • HCL AppScan CodeSweep • JMeter tests • JUnit tests • Performance tests • Postman resources • Virtual services
Viewing test run results on the Azure test plans dashboard	<p>When you use the HCL OneTest™ Studio Azure extension in Azure pipelines to run tests on HCL OneTest™ Server, you can view the test results on the Jobs page. If you want to view the results of the test runs on the test plans dashboard, you must set up test plans with test cases for your project in Azure. You must also provide details about the Azure server, your personal access token, the Azure project, the test case, and the test plan while you configure the test run for the test results to display on the test plans dashboard.</p>

What's New in 10.2.2

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.2.2.

Feature title	Description
Support to install HCL OneTest™ Server on OpenShift is removed	OpenShift is no longer supported as a platform on which you can install HCL OneTest™ Server 10.2.2.
Authoring tests on HCL OneTest™ Server	<p>You can now use the test authoring capability of HCL OneTest™ Server that utilizes a containerized version of the HCL OneTest™ Studio desktop authoring products to author tests and virtual services.</p> <p>You can also modify test assets or virtual service resources that you create by using the containerized products or the test assets or resources in the project repositories that you created in the desktop products and committed to the repositories.</p>

Feature title	Description
Support to add multiple repositories simultaneously	<p>Previously, you were able to add individual Git repositories one at a time to your project on HCL OneTest™ Server.</p> <p>Now, you can add multiple repositories simultaneously that are contained in the same repository domain.</p> <p>See Adding repositories to a server project on page 1014.</p>
Support to configure and run individual test assets	<p>Previously, you were able to configure and run Suites, Compound Tests, or Schedules on HCL OneTest™ Server. Now, you can configure and run individual test assets such as an API test, a Functional test, or Performance test.</p> <p>See Test run configurations on page 446.</p>
Support to run instances of virtual service resources on a remote Kubernetes cluster	<p>You can now configure and run instances of virtual service resources on a remote Kubernetes cluster. You can select the remote Kubernetes cluster as the location to run the virtual service resources after you set up and add the remote cluster to your project.</p> <p>See Configuring a run of a virtual service on page 750.</p>
Support to run test assets on remote agents or from a remote Kubernetes cluster	<p>You can now select remote agents or a remote Kubernetes cluster as a location to run the test assets. You can select the remote agent as the location to run the test asset after you set up and add the remote agent or the Kubernetes cluster to your project.</p>
Enhanced Jenkins plugins to run tests from a Jenkins pipeline project	<p>The HCL OneTest™ Server Jenkins plugins are enhanced so that you can now run tests as a job in a Jenkins Pipeline project.</p>
Configuring additional Jira fields	<p>Jira server administrator can now configure additional fields as Required fields in the project on the Jira server to be displayed in the Raise defect in Jira window while creating a defect from HCL OneTest™ Server.</p> <p>See Creating Jira issues for the test results on page 895.</p>
Collecting and viewing resource monitoring usage data	<p>You can create queries to retrieve monitoring usage data that is collected for a time period during a test run.</p>

Feature title	Description
	See Collecting and viewing resource monitoring usage data on page 883 , Configuring a test on page 884 , Viewing resource monitoring usage data on page 885 , and Creating queries for resource monitoring usage data on page 889 .
Viewing test logs	When a test run is complete, a test log report is generated for the tests which are configured to collect test logs data. The test log contains details about the events in the tests and their properties. See Test logs on page 808 .
Querying test log data	You can create queries to view test log data for all the test results in a project or for one particular result. See Test log queries on page 823 and Configuring and running log queries on page 824 .
Viewing results data in Grafana	By using HCL OneTest™ Server and Grafana, you can retrieve and explore test log data that is generated in HCL OneTest™ Server. See Test results data in Grafana on page 801 , Installing and configuring the DevOps Test Hub Results Data plugin on page 802 , and Viewing test results data in Grafana on page 804 .
Viewing test results in Surefire format reports	After you execute JMeter, JUnit, API, and Functional tests with advanced settings in HCL OneTest™ Server, you can view the test results in a Surefire format report by using dedicated URLs. See Viewing Surefire format reports on page 807 .

What's New in 10.2.1

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.2.1.

Feature title	Description
Integration with IBM Engineering Test Management	You can now integrate HCL OneTest™ Server with IBM® Engineering Test Management to run the test assets in a project in your team space from Engineering Test Management. After the tests run, you can view the results on the Results page. See Integration with IBM Engineering Test Management on page 967 .
Overriding results database	When you want to override the Results Database that is configured in the API Suite project, you can now provide the URL, user name, password, and the database driver

Feature title	Description
	as program arguments in the Advanced settings in the Execute test asset dialog box. You can optionally use an encrypted password that is generated by using the Encrypt-Password application that is in the installation directory of HCL OneTest™ API.
Running a scan of an application code	<p>You can use HCL OneTest™ Server to scan your application code, similar to using the AppScan CodeSweep plugin in Visual Studio. You must create a <code>.codesweep</code> file that contains information about the assets in the repository and the folders with the application code that must be scanned. You can optionally, specify the rules to ignore, folders with code to be excluded from the scan, or the parameters for a verdict of the scan in the file.</p> <p>After you commit the assets to a remote repository, you can add the repository to your project in a team space, and then run a scan of the assets on HCL OneTest™ Server.</p> <p>See Configuration of a scan of an application code on page 716.</p>
Running tests on remote Kubernetes clusters	<p>You can now run the supported test resources on remote Kubernetes clusters that you register with a team space on HCL OneTest™ Server and add the Kubernetes cluster to your project in the team space.</p> <p>You can select the remote cluster as a location to run the tests when you configure a run.</p> <p>You can edit the configurations of a cluster, remove a cluster from a project, or unregister the cluster from a team space.</p> <p>See Management of remote Kubernetes clusters on page 417.</p>

What's New in 10.2.0

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.2.0.

Feature title	Description
Installing server software on Azure Kubernetes Service	You can now install HCL OneTest™ Server on Azure Kubernetes Service that has a Kubernetes environment to run functional, integration, and performance tests. See Installation of DevOps Test Hub on Azure Kubernetes Service on page 94 .
Creating and maintaining multiple team spaces	Previously, HCL OneTest™ Server supported only one logical partition as the initial team space. Now, you can create and maintain multiple team spaces in HCL OneTest™

Feature title	Description
	<p>Server. The initial team space is available for all licensed users of HCL OneTest™ Server before you create any other team space.</p> <p>You can perform the following tasks in team spaces:</p> <ul style="list-style-type: none"> • Configure licenses or licensed users for each team space. • View all team spaces created on HCL OneTest™ Server and the team spaces that you can access from the Team Space Dashboard page. • Register Dockers, agents, intercepts, or Resource Monitoring agents with a team space before you use them in your projects of that team space. • Add team space members with the following roles: <ul style="list-style-type: none"> ◦ <i>Team Space Owner</i> ◦ <i>Project Creator</i> ◦ <i>Architect</i> ◦ <i>Member</i> • Isolate the projects from the users who are not members of the team space. <p>See Team space overview on page 168.</p>
Configure a repository with any default branch in a team space	<p>Now, you can configure any branch in a repository as the default branch in a team space.</p>
Associating test suites to the components of the System Model	<p>You can now associate test suites to the components of the System Model in addition to the virtual resources. After you associate the test suite, you can then click the test suite to view them on the Execution page.</p> <p>See Associating resources with components on page 359.</p>
Starting or stopping virtual services associated with components	<p>After you associate virtual services with components in the System Model page, you can then navigate to the Resources or Instances page to either start an instance or stop a running instance.</p> <p>See Viewing resources that are associated with components on page 361.</p>
Viewing and filtering test assets by their associated components on the Execution page	<p>The components in the System Model page that are associated with test assets can now be viewed in the Components column on the Execution page. You can also use the component as a parameter in a filter rule to obtain test assets associated with a specific component. You can return to the System Model page by clicking the component displayed for the test asset in the Components column on the Execution page.</p>

Feature title	Description
Viewing and stopping virtual service instances associated with components	<p>After you start an instance of the virtual services that are associated with components in the System Model page, you can then navigate to the Instances page to either view the running instance or stop a running instance. You can also view the components that are associated with running instances of virtual services from the Instances page.</p> <p>See Viewing running instances of virtual services on page 770.</p>
Introducing the Concurrent Virtual Services licenses	<p>Now, you need a separate Concurrent Virtual Services license to run an instance of the virtual services on HCL OneTest™ Server.</p> <p>See DevOps Test Hub licensing information on page 180.</p>
Viewing running instances associated with components	<p>After you have associated virtual services with components in the System Model and started instances, you can view the instances that are associated with the components from the Instances page. You can also filter the view to display running instances by using components as a criteria in the filters.</p>
Hiding inactive instances from the virtual service Instances page view	<p>You can now hide the virtual service instances that are stopped or those that failed, from the display on the virtual service Instances page via a single button.</p> <p>See Viewing running instances of virtual services on page 770.</p>
Stopping multiple running instances of virtual services	<p>You can now stop multiple or all running instances of virtual services from the Instances page.</p> <p>See Stopping virtual services on page 787.</p>
Navigating to the Progress page to view the state of test asset runs	<p>You can now navigate to the Progress page from the Execution page to view the runs that were started, scheduled, or completed by clicking the Show in the Progress page icon that is enabled for the specific test asset.</p> <p>See Viewing the state of test assets on page 727.</p>
Stopping or Canceling multiple test runs	<p>You can now stop multiple or all running test assets from the Progress page. You can also cancel multiple or all scheduled runs of test assets from the Progress page.</p> <p>See Stopping test runs on page 736 or Canceling scheduled test runs on page 739.</p>

Feature title	Description
Defining resource monitoring sources for a team space	As a team space owner, you can define resource monitoring sources in the team space so that they can be shared between multiple projects in the team space. During a performance schedule execution with resource monitoring labels, the sources defined in a team space are collected with the sources that are defined in a project whenever they match the provided labels. As a user of a team space, you can use the resource monitoring sources that are defined in the team space.
Agents listed for selection based on capabilities	When you install static agents and add the agents to your project on HCL OneTest™ Server, the agents based on the capabilities that match with the agent specified in the test assets are listed for an override. The agents with the best matching capabilities are listed on the top followed by other agents with mismatched capabilities.
Adding categories and properties to user-defined capabilities	<p>The user-defined capabilities feature is enhanced. Now, you can group user-defined capabilities under categories that you create. You can also add name-value pairs as properties for the capabilities.</p> <p>See Working with the capabilities of a performance agent on page 391.</p>
Establishing relationship among schemas based on referential integrity	Previously, you were able to only view the relationships among JDBC-supported schemas on the Schema Canvas page. You can now establish parent-child relationships among multiple non-JDBC schemas on the Schema Canvas page.
Generating DDL statements for multiple schemas	You can now generate the Data Definition Language (DDL) statements for single or multiple non-JDBC schemas. You can use the DDL statements to create a table in your database. Each job of generating DDL statements for the selected schemas generates a SQL file.
Generating the test data by using multiple schemas	<p>Previously, you were able to generate the test data for only multiple JDBC-supported schemas. You can now generate the test data for multiple non-JDBC schemas by maintaining referential integrity. You can also modify the relationships among non-JDBC schemas on the Schema Canvas page.</p> <p>You can view the relationship among the selected schemas on the Schema Canvas page.</p>

What's New in 10.1.3

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.1.3.

Feature title	Description
<p>Introducing Team Spaces on HCL OneTest™ Server</p>	<p>HCL OneTest™ Server is now provisioned to display an initial Team Space, which is a logical partition of HCL OneTest™ Server.</p> <p>The operations that you managed at the server level can now be performed at the Team Space level. You can now configure licenses or an SMTP server, create projects, register the remote agents, dockers, or intercepts within a Team Space. See Team space overview on page 168.</p>
<p>System modeling</p>	<p>You can create a system model to view the logical representation of any application under test in a Team Space. The system modeling is a Tech Preview feature.</p> <p>You can configure a Team Space repository, which is different from the project repository, to save the system model.</p> <p>You can create components that represent different types of assets and associate the components by defining the relationship between the components. You can also create child components for components. You can use these concepts to build a system model and view the model. You can then publish the system model to the Team Space repository.</p> <p>You can also associate the virtual services in a project repository with components of the system model. From the associated component, you can view and run the virtual services.</p> <p>See System modeling on page 339.</p>
<p>Working with virtual services</p>	<p>You can now view the virtual service resources that are in your project repositories from the Resources page and then configure a run of the virtual service. Working with virtual services is a Tech Preview feature.</p> <p>After you start a virtual service, the running instances can be viewed from both the Resources page and Instances page.</p> <p>You can perform the following tasks on virtual services:</p> <ul style="list-style-type: none"> • View information about the details, behavior, usage, routing rules, and diagnostic information of the virtual service instance. • Modify the behavior or the logging level of a running instance. <p>See Management of virtualized services on page 741.</p>

Feature title	Description
Stubs are excluded from Execution and Progress pages	<p>Starting from 10.1.3, stubs are not displayed on the Execution page to start them nor are they displayed on the Progress page after they are started. Stubs must be started from the Virtualization > Resources page and stubs that are running can be viewed from the Virtualization > Instances page.</p> <p>See Management of virtualized services on page 741.</p>
Running JUnit tests on HCL OneTest™ Server	<p>You can now run JUnit tests that are within a Maven project from HCL OneTest™ Server.</p> <p>You must commit the Maven project as a <code>pom.xml</code> to a repository. After you add the repository to your project on HCL OneTest™ Server, you can configure a run of the JUnit tests from the Execution page.</p> <p>After the tests are run, you can view the detailed reports of the tests. See Configuring a run of a JUnit test on page 674.</p>
Agents are selected automatically for an override	<p>When you install static agents V10.1.2 and add the agents to your project on HCL OneTest™ Server, HCL OneTest™ Server determines and then displays the capabilities of the agents. Further, HCL OneTest™ Server compares the capabilities of the agents with the required capabilities specified for the agent in the test assets and automatically selects the most suitable agent for an override. You can either proceed with running the test on the agent selected or select an agent from the agents displayed that have the same capabilities. See Adding a performance agent to a project on page 389.</p>
Working with capabilities that you define for static agents	<p>When you install static agents V10.1.3 and register them with HCL OneTest™ Server, you can add, view, edit, or delete the capabilities of static agents. See Working with the capabilities of a performance agent on page 391.</p>
Refreshing of the migrated project repositories	<p>After you migrate a project from a previous version to a later version of HCL OneTest™ Server, the repositories that are configured for the project are refreshed only after the project owner logs in to HCL OneTest™ Server.</p>
Generating the test data by using multiple schemas	<p>Previously, when you used the JDBC connection to generate the test data by using HCL® OneTest™ Data, then you were able to generate the test data for only one schema. You can now generate the test data for multiple schemas simultaneously by maintaining referential integrity. Multiple schemas that you use to generate the test data now belong to a single JDBC connection.</p>

Feature title	Description
	You can also view the relationship among the selected multiple schemas on the Schema integrity page.
Execution of properties based on priority	When you set multiple properties for any item type, then HCL® OneTest™ Data uses an order of priority while executing the properties.
Establishing a secure and trusted integration between applications and HCL® OneTest™ Data	You can now establish a secure and trusted integration between applications and HCL® OneTest™ Data by placing the SSL certificate in the trust store.

What's New in 10.1.2

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.1.2.

Feature title	Description
Documentation	You can now quickly add your comments to provide your feedback about the documentation for HCL OneTest™ Server by posting to a product forum. To add your comments as feedback, you must first register at HCL Support. Your login information along with your comments are governed by the privacy policy of HCL. By commenting, you accept the terms of use of HCL.
Installation	You can now install HCL OneTest™ Server on Ubuntu (using k3s) platform. See Installing DevOps Test Hub on Ubuntu on page 109 .
Enhanced license configuration settings	If you are an administrator, the Configure license option enables you to change the configuration values such as the URL and ID of the License Server. An administrator can also add, remove, or replace named users and enable or disable the automatic self-acquisition of the named user licenses.
Refreshing repositories	The repositories are refreshed automatically at regular intervals to ensure that each repository is in synchronization with the remote repository. You can also trigger the refresh activity manually for a repository to fetch the changes instantly. See Refreshing repositories manually on page 1031 .
Creating webhooks	You can now create a webhook for a repository in Git server. For each push event in the Git server, the webhook notifies HCL OneTest™ Server to trigger the refresh ac-

Feature title	Description
	<p>tivity to pull the latest changes to the server repository. See Creating webhooks on page 1034.</p>
<p>Running Postman test collections</p>	<p>You can now run tests that you create in Postman on HCL OneTest™ Server. You must export the collections and environments to files that you commit to a repository. You must add the repository to your project on HCL OneTest™ Server.</p> <p>Before you run the Postman tests, you must ensure to enable the Postman microservice.</p> <p>You can configure a run of the entire collection or any of its subfolders from the Execution page. While you configure the run, you can provide additional command options as you did in the Newman CLI. After the tests are run, you can view the detailed reports of the tests. See Configuring a run of a Postman test on page 683.</p>
<p>Running JMeter tests on HCL OneTest™ Server</p>	<p>You can now run tests the JMeter tests that you create in Apache JMeter on HCL OneTest™ Server. Before you run the JMeter tests, you must enable the JMeter microservice on HCL OneTest™ Server.</p> <p>You must package the JMeter tests along with other associated assets such as properties files, certificates, library jars, and dependencies as a project. The project must contain the project file (<code>jmeterRoot.jprj</code>). The tests and the other assets must be contained in subdirectories within the project directory.</p> <p>You must commit the JMeter project to a repository. After you add the repository to your project on HCL OneTest™ Server, you can configure a run of the JMeter tests from the Execution page.</p> <p>After the tests are run, you can view the detailed reports of the tests.</p> <p>See Configuring a run of a JMeter test on page 665.</p>
<p>Viewing capabilities of registered agents</p>	<p>Previously, you were able to view the details about the agents. You can now view and ascertain the system and application capabilities of all agents that are registered with HCL OneTest™ Server. With the information presented, you can make informed decisions about selecting an agent for running tests from HCL OneTest™ Server. See Adding a performance agent to a project on page 389.</p>
<p>Managing notifications</p>	<p>You can now choose whether to receive in-app, email, or both notifications for each notification channel. By default in-app notifications are enabled. The notifications are</p>

Feature title	Description
	about tasks that need completion from users or for information purposes. See Managing email notifications on page 1047 .
Configuring notifications	As an administrator, you can now set up an SMTP email server to send email notifications. The notifications are about tasks that need completion from the users or for information purposes. See Configuring an SMTP server to manage email notifications on page 1050 .
Support for integration with Azure	You can now use the HCL OneTest™ Studio extension to establish a trusted and secure connection between Azure DevOps Server and HCL OneTest™ Server to run the HCL OneTest™ Server tests from the Azure DevOps Server. See Integration with Azure DevOps on page 926 .
Enhanced the integration with HCL® Launch	You can now use the HCL OneTest™ Server Launch plugin to establish a trusted and secure connection between the HCL® Launch server and HCL OneTest™ Server to run the HCL OneTest™ Server tests from the HCL® Launch server.
Support for integration with Jenkins	You can now use the HCL OneTest™ Server Jenkins plugin to establish a trusted and secure connection between the Jenkins server and HCL OneTest™ Server to run the HCL OneTest™ Server tests from the Jenkins server. See Integration with Jenkins on page 973 .
Support for integration with Jira	HCL OneTest™ Server is now integrated with Jira software, as a result, you can create a defect in Jira from the HCL OneTest™ Server UI. See Configuration of Atlassian Jira as a change management system on page 892 .
Importing a CSV file with a separator character	You can now import a CSV file that has a separator character into the dataset and configure the required separator values in the CSV Editor. See Editing a dataset on page 226 .
HCL® OneTest™ Data integration with HCL® Launch	You can now integrate HCL® OneTest™ Data with HCL® Launch by using the HCL® OneTest™ Data Launch plugin to generate the test data on the HCL® Launch server.
Setting regular expressions as a basic property	Previously, the Regular Expression field was available only under Advanced properties . You can now also find this field as a basic property of an item type in the Properties dialog box.

Feature title	Description
Managing HCL® OneTest™ Data project	You can now archive or delete the project, if you no longer require the data design environment that you created in your HCL® OneTest™ Data project. You can also unarchive any of your archived projects.

What's New in 10.1.1

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.1.1.

Feature title	Description
Reviewing system requirements	You can now find system requirements information about hardware and software compatibility in this documentation. For more information, see System Requirements for DevOps Test Hub 2024.03 (11.0.1) on page 79 .
Upgrading the server software	You can now upgrade to the latest version of HCL OneTest™ Server on the Red Hat OpenShift platform by using the helm upgrade command from 10.1.0. See DevOps Test Hub upgrade methods on page 114 .
Support for running Compound Tests that contain mobile tests	You can now configure a run for a Compound Test that contains mobile tests and then run the Compound Test on HCL OneTest™ Server. See Configuring a run of a Compound Test that contains mobile tests to run by using variables on page 509 .
Overriding Resource Monitoring settings in a performance schedule	Now in the Execution dialog, the Resource Monitoring tab displays the settings related to the Resource Monitoring sources and labels as they were set in the performance schedule. This information is helpful to decide whether you want to override them with the Resource Monitoring sources that are available in your Server project, or not. See Controlling resource monitoring sources in a schedule on page 878 .
HCL OneTest™ Server integration with HCL® Launch	You can now integrate HCL OneTest™ Server with HCL® Launch by using the HCL OneTest™ Server Launch plugin to run tests on the HCL® Launch server.
Service virtualization: Tech Preview Virtualizing a service in a namespace of HCL OneTest™ Server	Previously, you specified the short name of a service that you wanted to virtualize as the host for the HTTP transport and specified the namespace in the recording settings for an Istio proxy. For example, to record the reviews service that had an FQDN of <code>reviews.default.svc.cluster.local</code> the host for the HTTP transport to be specified was <code>reviews</code> , and the namespace to be specified for the Istio Proxy was <code>default</code> . However, Istio routing in HCL OneTest™ Server only uses the host name and port to generate the

Feature title	Description
	<p>routing rule and for stubbing. Therefore, the host name must contain the namespace so that HCL OneTest™ Server can correctly identify the service that is being stubbed.</p> <p>Now, you can enable HCL OneTest™ Server to use the namespaces that you want by providing the namespaces at the time of the installation of the server software.</p>
Managing test data generated jobs and API history	<p>HCL® OneTest™ Data now periodically deletes all the generated test data and the jobs of test data generation with API history to clear the memory. After changing the default configuration settings, you can retain the test data generated jobs and the API history of the jobs for a specific period.</p>
Generating test data for MongoDB	<p>You can now generate the test data for the MongoDB by using HCL® OneTest™ Data. HCL® OneTest™ Data generates the test data, and then directly writes the generated test data in MongoDB.</p>
Generating test data for Excel	<p>You can now generate the test data for an Excel file by using HCL® OneTest™ Data. You can then download the test data on your computer.</p>
Generating test data by using the pairwise data generation method	<p>You can now generate an optimum test data that covers all possible combinations of the values of the item types for each pair of item types in a schema by using the pairwise data generation method.</p>
Copying artifacts of a schema to another schema	<p>You can now copy artifacts of a schema to a new or an existing schema.</p>
Performing operations by using REST APIs in HCL® OneTest™ Data	<p>You can now perform operations on the schemas in HCL® OneTest™ Data by using a set of REST APIs from any external application.</p>
Saving a copied dataset	<p>You can now save a copy of a dataset in a different folder or make a copy with a different name, by using the Save As option.</p> <p>See Working with Datasets on DevOps Test Hub on page 216.</p>
Displaying icons to view who is editing datasets	<p>In HCL OneTest™ Server, you can now see all members who are editing a particular dataset.</p> <p>See Editing a dataset on page 226.</p>

What's New in 10.1.0

The following section lists the features, enhancements, or other changes made in HCL OneTest™ Server 10.1.0.

Feature title	Description
Installation	<p>HCL OneTest™ Server can only be installed on the following platforms:</p> <ul style="list-style-type: none"> • Red Hat OpenShift 4.2+ • Ubuntu (using MicroK8s) <p>See Installation of DevOps Test Hub on page 93.</p>
Server Licensing	<p>You can purchase named user licenses for HCL OneTest™ Server and install the licenses on the FlexNet Operations server.</p> <p>For more information, see HCL license portal on page 178.</p>
Backup and restore of user data	<p>You can back up and restore the user data in HCL OneTest™ Server V10.1.0. You can also migrate the backed-up user data from the previous versions of HCL OneTest™ Server to HCL OneTest™ Server V10.1.0.</p> <p>For more information, see Backup and restoration of the DevOps Test Hub data on page 125.</p>
Running AFT Suites on remote agents	<p>You can now select an agent added to your project on HCL OneTest™ Server to override the agent configured in the AFT XML file in the test asset.</p>
Selecting test resources	<p>You can now select either the current or previous versions of the test resources that you want to run while you are configuring the test run.</p>
Resetting test run configurations	<p>You can now reset the settings you configured for a test run when you are either initiating or repeating the test run. For more information, see Resetting the configuration settings for a test run on page 735.</p>
Support for running an HTML test in a Compound Test on HCL OneTest™ Server	<p>You can now, run an HTML test as a compound test on HCL OneTest™ Server and choose to override the configured browser while configuring the test.</p> <p>For more information, see Configuring a run of a Compound Test that contains HTML tests on page 498.</p>

Feature title	Description
Navigating from a result to the associated Jaeger trace	<p>Now, test logs are delivered in a Jaeger format by default for executed compound tests and schedules. You can still set a Program Argument to re-enable the traditional test log format when you run the tests.</p> <p>For more information, see Test results and reports on page 789.</p>
Relaunching an execution from the Results view	<p>From the Results view, you can re-execute a test with the same parameters as the ones that you previously set for the previous test run. You can also re-execute multiple tests at a time with the same parameters as the initial ones.</p> <p>For more information, see Re-executing tests from results on page 799.</p>
Controlling Resource Monitoring sources	<p>Now you can use label matching to control what Resource Monitoring sources are collected during the execution of a Performance schedule.</p> <p>For more information, see Controlling resource monitoring sources in a schedule on page 878.</p>
Functional Test reports	<p>A Unified Report is generated as a result of a functional test execution.</p> <p>For more information, see Test results and reports on page 789.</p>
Querying metrics	<p>Now the Resource Monitoring service can query metrics from a target monitored by a running Prometheus server with default or customized queries.</p> <p>For more information, see Monitoring metrics collected by a Prometheus server on page 851.</p>
Scrapping metrics	<p>The Resource Monitoring service can scrap metrics exposed by an OpenMetrics exporter through metric counters.</p> <p>For more information, see Monitoring metrics exposed by an OpenMetrics exporter on page 855.</p>
Selecting branches	<p>In earlier versions, branch selection was configured on the Repositories page and applied globally to all users. Now, each user is presented with a Branch field that lists all the branches of the repositories added to your project. The list is available in the Execution, Datasets and Overview pages HCL OneTest™ Server. For more information, see Selecting the global branch in a project on page 1033.</p>

Feature title	Description
Creating secret collections	<p>Members with the Tester role assigned in a project, can now create secrets collections.</p> <p>For more information, see Protecting API test assets by using secrets on page 1035.</p>
Project repository	<p>You can now add a repository with multiple folders to a single project. With this improvement, each project is now restricted to a single copy of a given repository.</p>
HCL OneTest™ Server integration with Azure DevOps integration with Azure DevOps	<p>You can now integrate HCL OneTest™ Server with Azure DevOps by using the HCL OneTest™ Studio extension to run tests as tasks in Azure DevOps pipelines.</p> <p>For more information, see Integration with Azure DevOps on page 926.</p>
Running stubs	<p>You can now run stubs that you create in HCL OneTest™ API, from the Execution page on HCL OneTest™ Server. You can run IBM® WebSphere® MQ stubs and HTTP stubs on HCL OneTest™ Server.</p> <p>For more information, see Configuring a run of a virtual service on page 750.</p>
Running stubs for Istio services	<p>After you create stubs for Istio in HCL OneTest™ API that virtualize services in a Kubernetes cluster, you can run those stubs on HCL OneTest™ Server. The stubs can virtualize services that use the Istio service mesh for the following types of requests:</p> <ul style="list-style-type: none"> • Requests received by services in the Istio service mesh. • Requests sent from namespaces in the Istio service mesh to external services that are not in the Istio service mesh.
Stub usage statistics	<p>After you run stubs on HCL OneTest™ Server, you can view the stub usage statistics from the Stub Usage page. For more information, see Viewing usage statistics of virtual services on page 784.</p>
Routing rules	<p>You can view the details of the intercepts, proxies, or stubs with routing rules defined that are connected to HCL OneTest™ Server, from the Routing Rules page. For more information, see Viewing routing rules of the virtual services on page 783.</p>
Using Excel Data Sources	<p>Excel Data Sources defined in HCL OneTest™ API projects can now be used as Datasets in your HCL OneTest™ Server projects. Note that Excel Data Sources that define a specific worksheet to use are ignored.</p>

Feature title	Description
HCL® OneTest™ Data	HCL® OneTest™ Data is a test data generation tool. You can use HCL® OneTest™ Data to generate the sample test data to perform tests. This tool is now available as a component of HCL OneTest™ Server.
Authoring datasets	Now, you can create, edit, or delete a dataset in HCL OneTest™ Server. You can also publish the created dataset to Git repository so that other members of the project can use your dataset in their test asset run. For more information, see Working with Datasets on DevOps Test Hub on page 216 .

Deprecations and removals

You can find the features or capabilities announced as deprecated and the features or capabilities that are removed in HCL DevOps Test Hub (Test Hub). DevOps Test Hub is the new name of HCL OneTest™ Server from 11.0.0 onwards.

When features or capabilities are announced as deprecated, they continue to be supported although such features or capabilities are no longer enhanced and might be removed in a subsequent release. If you use a deprecated feature, you must migrate to the alternate feature, if applicable, before the deprecated feature is removed.

The removed features will not be available in the product version in which they are removed.



Note: You can contact the HCL Client Support team if you have any concerns.

You can find information about the following deprecations or removals:

- [Product features or capabilities on page 38](#)
- [Dependent software on page 39](#)

Product features or capabilities

You can find information about the features or capabilities of Test Hub that are announced as deprecated in an earlier version, and the version in which they are removed.



Note: The features removed in this version are listed first, followed by the features deprecated. The features removed or deprecated in earlier versions are listed later in the table.

Feature or capability	Feature deprecated in version	Alternate feature, if applicable	Feature removed in version
The test authoring capability of HCL OneTest™ Server that utilizes a containerized version of the HCL OneTest™ Studio desktop authoring products to author tests and virtual services.	10.5.0	None	11.0.0
Support to run Istio based virtual services	10.5.0		11.0.0
Use of Jaeger to display test logs.	10.2.3	The capability provided by Jaeger to provide the test log traces is replaced with the use of the built-in results service to provide test logs.	11.0.0
Support of Prometheus for recording usage data of virtual services		The capability provided by Prometheus is replaced with the use of the built-in logging capability for virtual services.	10.5.3

Dependent software

You can find information about the dependent software that were announced as deprecated in an earlier version and the version in which they were removed. Dependent software can include the operating systems, license servers, applications, or software required for using Test Hub.



Note: The features removed in this version are listed first, followed by the features deprecated. The features removed or deprecated in earlier versions are listed later in the table.

Dependent software	Deprecated in version	Alternate OS or software, if applicable	Support removed in version
HCL Local License Server 2.0 (also known as 2021.05)	Not applicable	HCL Local License Server 3.0 (also known as 2022.12)	10.5.3

Defect fixes

You can find the defects that are fixed in this version of HCL DevOps Test Hub (Test Hub). You can also find the defects that were fixed in the earlier versions of Test Hub. DevOps Test Hub is the new name of HCL OneTest™ Server from 11.0.0 onwards.

You can find the following information:

- [Defects fixed in DevOps Test Hub 2024.03 \(11.0.1\) on page 40](#)
- [Defects fixed in earlier versions on page 40](#)

Defects fixed in DevOps Test Hub 2024.03 (11.0.1)

The defects fixed in Test Hub 2024.03 (11.0.1) are as follows:

Table 1. List of fixes in Test Hub 2024.03 (11.0.1)

ID	Description
TP-91596	Previously, when you ran a Web UI test with a custom code to use JDBC libraries, Test Hub displayed <code>java.lang.UnsupportedClassVersionError</code> . This problem is fixed.

Defects fixed in earlier versions

You can find information about the defects fixed in each of the following versions:

Defects fixed in 11.0.0

The defects fixed in Test Hub 11.0.0 are as follows:

Table 2. List of fixes in Test Hub

ID	Description
TP-89626	Previously, the terminated Execution pods incorrectly displayed the status as an error when the API test or API Suite returned the result as <i>Fail</i> . This problem is fixed.
TP-88262	Previously, when you ran Web UI tests that had a large number of test steps, the unified report failed to be exported as a PDF. This problem is solved.
TP-89553	Previously, the unified report was displayed incorrectly when you ran a Compound Test that contained mobile tests, which ran on either the devices, emulators, or simula-

Table 2. List of fixes in Test Hub (continued)

ID	Description
	<p>tors that were connected to an agent computer or the devices that were connected to a mobile device cloud.</p> <p>This problem is fixed.</p>

Defects fixed in 10.5.4

The defects fixed in HCL OneTest™ Server 10.5.4 are as follows:

Table 3. List of fixes in 10.5.4

ID	Description
TP-89167	<p>Previously, when you exported a statistic report for a test result, the downloaded report displayed the Agent Health report as the default report.</p> <p>This problem is fixed.</p> <p>The downloaded report now displays the UI Statistical report as the default report.</p>
TP-86087	<p>Previously, when you started a large number of test assets simultaneously, you were required to determine the capacity of the system, size of projects, and adjust the rate accordingly.</p> <p>This problem is fixed.</p>

Defects fixed in 10.5.3

The defects fixed in HCL OneTest™ Server 10.5.3 are as follows:

Table 4. List of fixes in 10.5.3

ID	Description
TP-88119	<p>Previously, if you set the Helm option gateway.allowOrigin during installation of HCL OneTest™ Server, it was mandatory for you to include the server URL in the string.</p> <p>This problem is fixed.</p> <p>Now, you need to only include strings for additional servers and you do not have to include the server URL string in the gateway.allowOrigin option.</p>

Table 4. List of fixes in 10.5.3 (continued)

ID	Description
TP-88111	<p>Previously, the default configuration in Keycloak incorrectly assigned all users of HCL OneTest™ Server the privileges of a server administrator.</p> <p>This problem is fixed.</p>
TP-88079	<p>Previously, when members of a team space entered the URL of a page in a project, they were incorrectly provided the access to view the page even though they were not a member of that project.</p> <p>This problem is fixed.</p>
TP-86646	<p>Previously, when filtering the virtual service instances that are listed on the Instances page by activity, HCL OneTest™ Server incorrectly filtered by overall activity rather than recent activity.</p> <p>This problem is fixed.</p>
TP-86105	<p>Previously, the project overview page did not display the statistics of the test results that were published to HCL OneTest™ Server from any of the desktop products.</p> <p>This problem is fixed.</p> <p>Now, if a project has no configured repositories, the project overview page displays the statistics of the test results published to HCL OneTest™ Server from any of the desktop products.</p>

Defects fixed in 10.5.2

The defects fixed in HCL OneTest™ Server 10.5.2 are as follows:

Table 5. List of fixes in 10.5.2

ID	Description
TP-86049	<p>Previously, when you entered non-alphanumeric characters in search boxes on any page on HCL OneTest™ Server, the search resulted in errors.</p> <p>This problem is fixed.</p>

Defects fixed in 10.5.1

The defects fixed in HCL OneTest™ Server 10.5.1 are as follows:

Table 6. List of fixes in 10.5.1

ID	
TP-84738	<p>When you associated Suites and tests with components in the system model in HCL OneTest™ Server 10.5.0, the associated components were not displayed under the Components column on the Execution and Results pages.</p> <p>This problem is fixed.</p>
TP-84688	<p>Previously, when you deleted multiple associations to test resources that were in different projects, HCL OneTest™ Server failed to delete all the associations in the system model.</p> <p>This problem is fixed.</p>
TP-84675	<p>Previously, when you added a static agent to multiple projects in HCL OneTest™ Server, the agent was not added to all the projects and failed after a few projects were added.</p> <p>This problem is fixed.</p>
TP-84492	<p>Previously in Keycloak, when using the LDAP user provided certificates were not imported into the Keycloak truststore that resulted in errors when authenticating.</p> <p>This problem is fixed.</p>

Defects fixed in 10.5.0

The defects fixed in HCL OneTest™ Server 10.5.0 are as follows:

Table 7. List of fixes in 10.5.0

ID	Description
TP-84022	<p>Previously, when you ran a Web UI test with an associated dataset, the test failed to read all the rows and stopped after completing a few iterations.</p> <p>This problem is fixed.</p>

Table 7. List of fixes in 10.5.0 (continued)

ID	Description
TP-83686	A Sign-Up link is no longer offered on the login page by default to improve security posture. It can be enable during install or afterward from Keycloak by enabling Login > User Registration .
TP-82448	Previously, when you ran a performance test that contained integration tests on HCL OneTest™ Server, the hierarchy of test iteration in the test log was incorrect. Therefore, the integration test steps in the test log were displayed under the first activity that was captured for the performance test. This problem is fixed.
TP-76020	Previously in Keycloak, when using the LDAP user provided certificates were not imported into the Keycloak truststore that resulted in errors when authenticating. This problem is fixed.

Defects fixed in 10.2.3

The defects fixed in HCL OneTest™ Server 10.2.3 are as follows:

Table 8. List of fixes in 10.2.3

ID	Description
TP-81758	Previously, if you specified a secrets collection while configuring a run of a virtual service, the specified collection was ignored and an attempt was made to use a secrets collection with the same name as the environment. This problem is fixed.
TP-81035	Previously, the default configuration of HCL OneTest™ Server had access to requests from any domain. Therefore, HCL OneTest™ Server was exposed to cross-origin resource sharing (CORS) attacks. This problem is fixed.
TP-81017	Previously, if you attempted to edit an API Suite that was in the git branch of your server project, the containerized API editing session was not launched because of non-omitted spaces in the path of the asset.

Table 8. List of fixes in 10.2.3 (continued)

ID	Description
	This problem is fixed.
TP-80637	<p>Previously, while you installed HCL OneTest™ Server on Azure Kubernetes Service (AKS) if the postgresql-init job did not complete early, it might have remained in a Pending state due to insufficient CPU resources.</p> <p>This problem is fixed.</p>
TP-80565	<p>Previously, when you deployed HCL OneTest™ Server on Azure Kubernetes Service (AKS), and if you did not set up the Domain Name Server (DNS) resolution of the client, you were unable to open a new editor session.</p> <p>This problem is fixed.</p>
TP-80321	<p>Previously, when you ran a performance or functional integration test on HCL OneTest™ Server that had the API test type as a RAML definition, the test failed to run.</p> <p>This problem is fixed.</p>

Defects fixed in 10.2.2

The defects fixed in HCL OneTest™ Server 10.2.2 are as follows:

Table 9. List of fixes in 10.2.2

ID	Description
TP-79765	<p>Previously, if two instances of a virtual service resource were run simultaneously with different environments and later stopped, the virtual services displayed their status as <i>STOPPED</i> but the virtual services might be left running in the container.</p> <p>This problem is fixed.</p>
TP-79322	<p>Previously, when a virtual service on the server was running and the server execution service restarted, it resulted in the loss of the routing rules for the virtual service.</p> <p>This problem is fixed.</p>
TP-79272	<p>Previously, when you started an HTTP virtual service by using the REST API and supplied an environment name where the name differed from the desktop project value in its letter-cas-</p>

Table 9. List of fixes in 10.2.2 (continued)

ID	Description
	<p>ing, for example, <i>env</i> compared to <i>Env</i>, the virtual service might have started but the routing rules were not created.</p> <p>This problem is fixed.</p>
TP-79218	<p>Previously, if you attempted to use the dataset override when you started an instance of a virtual services resource, you might have noticed that the dataset override did not take place.</p> <p>This problem is fixed.</p>
TP-79185	<p>When you run two virtual services concurrently from HCL OneTest™ Server, where the virtual services come from the same desktop project but are run with different environments, traffic might fail to be routed to one of the virtual service and HTTP 502 errors might be encountered.</p> <p>This problem is fixed.</p>
TP-78842	<p>Previously, when you ran a test on a remote Kubernetes cluster from a HCL OneTest™ Server installation that was configured with a non-default ingress secret name, the remote execution pod failed to start.</p> <p>This problem is fixed.</p>
TP-78841	<p>Previously, when you scheduled a test to run on a remote agent associated with your project in your team space, the test run failed to run and an error message was displayed.</p> <p>This problem is fixed.</p>
TP-78219	<p>Previously, when you updated the name of the project in Jira, and if you have used that project name on HCL OneTest™ Server, you were unable to configure the updated project name in the CHANGE MANAGEMENT SYSTEM tab of HCL OneTest™ Server.</p> <p>This problem is fixed.</p>
TP-78109	<p>Previously, when you stopped a large number of virtual service instances, you might have seen a 504 error although the virtual service instances were stopped successfully.</p> <p>This problem is fixed.</p>

Defects fixed in 10.2.1

The defects fixed in HCL OneTest™ Server 10.2.1 are as follows:

Table 10. List of fixes in 10.2.1

ID	Description
TP-77049	<p>Previously, when you ran the <code>reconcile-secrets.sh</code> script to update the password seed, the script displayed an error message.</p> <p>This problem is fixed.</p>
TP-76098	<p>Previously, during the deployment of HCL OneTest™ Server V10.2.0 on Azure Kubernetes Service (AKS) V1.20 or later, the deployment was failing with an error message.</p> <p>This problem is fixed.</p>
TP-74534	<p>Previously, when you tried to publish a created dataset after the migration from an earlier version to 10.1.3 or 10.2.0, you were unable to publish the dataset to a Git repository.</p> <p>This problem is fixed.</p>
TP-73109	<p>Previously the command-line interface was displaying warnings about Ingress deprecation during the installation of HCL OneTest™ Server 10.1.3 or 10.2.0 on Lightweight Kubernetes (k3s).</p> <p>This problem is fixed.</p>

Defects fixed in 10.2.0

The defects fixed in HCL OneTest™ Server 10.2.0 are as follows:

Table 11. List of fixes in 10.2.0

ID	Description
TP-71950	<p>Previously, when the server was under high load, in some cases, the results microservice restarted indefinitely. This issue occurred when the Kubernetes supervisor stopped the results microservice during a database locking process.</p> <p>This problem is fixed.</p>

Table 11. List of fixes in 10.2.0 (continued)

ID	Description
TP-71416	<p>Previously, when you attempted to run an API Suite on HCL OneTest™ Server that is installed on OpenShift, and when the API project contains a vast set of resources, the Execute button in the Execute test asset dialog box was enabled after a delay or at times not enabled at all.</p> <p>This problem is fixed.</p>
TP-70940	<p>Previously, during the installation of HCL OneTest™ Server, checks were not made to verify that the supplied ingress domain was valid.</p> <p>This problem is fixed.</p> <p>Now, when you provide the details for the ingress domain, checks are performed for the following constraints:</p> <ul style="list-style-type: none"> • The domain name must not exceed 253 characters in length. • Each element of the domain must not exceed 63 characters. <p>An additional check is made to ensure that the pattern, derived from the ingress domain, used to create hostnames dynamically, generates hostnames that conform to the same constraints. You can disable the additional check by specifying --set preflight.validateHostPattern=false. If the ingress domain name does conform to the constraints, some of the service virtualization features might not work correctly.</p>

Defects fixed in 10.1.3

The defects fixed in HCL OneTest™ Server 10.1.3 are as follows:

Table 12. List of fixes in 10.1.3

ID	Description
TP-72278	<p>Previously, when you attempted to run an API Suite on HCL OneTest™ Server that is installed on OpenShift, and when the API project contained a vast set of resources, the Execute button in the Execute test asset dialog box was enabled after a delay.</p> <p>This problem is fixed.</p>
TP-58061	<p>Previously, when you shut down HCL OneTest™ Server, the Suites, Compound Tests, or Schedules that were running did not stop.</p> <p>This problem is fixed.</p>

Defects fixed in 10.1.2

The defects fixed in HCL OneTest™ Server 10.1.2 are as follows:

Table 13. List of fixes in 10.1.2

ID	Description
TP-70414	<p>Previously, when you installed HCL OneTest™ Server V10.1.1 on an OpenShift platform where OpenShift Container Storage (OCS) was not available (For example, the IBM Cloud) then test asset runs failed.</p> <p>This problem is fixed.</p>
TP-69452	<p>Previously, when you started stubs on HCL OneTest™ Server, occasionally the stubs remained in transition and did not run.</p> <p>This problem is fixed.</p>
TP-67330	<p>In HCL OneTest™ Server V10.1.2, Kafka is replaced with RabbitMQ.</p> <p>Now you do not see a restart of the Kafka pod and some of the other microservices when the server is in an idle state.</p>
TP-67306	<p>Previously, from the Execution tab, you could not use the keyboard to search and select a branch.</p> <p>This problem is fixed.</p>
TP-66731	<p>Previously, you might have experienced errors when you published large reports, either from desktop products or when you executed a test on the server.</p> <p>This problem is fixed.</p> <p>You no longer have to set the options recommended as a workaround to publish large reports.</p>
TP-62379	<p>If you (as a user) requested to join an orphaned project, you saw a 404 not found error. You now see the following message: <code>You can't request to join a project that has no owners.</code></p>
TP-56453	<p>After about 20 minutes of an inactive browser session for HCL OneTest™ Server, if you attempted to use the existing browser session, you had to authenticate again by providing your username and password.</p>

Table 13. List of fixes in 10.1.2 (continued)

ID	Description
	This problem is fixed.

Defects fixed in 10.1.1

The defects fixed in HCL OneTest™ Server 10.1.1 are as follows:

Table 14. List of fixes in 10.1.1

ID	Description
TP-68740	This version addresses a problem with datapool encrypted values being logged as clear text to Jaeger log.
TP-67208	Previously, when you used the Undo or Redo options for a change in a dataset by using the CSV editor, the Save or Discard options were grayed out, thus you could not save or discard the changes. This problem is fixed.
TP-67032	Previously, before you upgraded to a new version, you had to cancel any scheduled execution. This problem is fixed.
TP-60737	Previously, GIT repositories that contained projects at the root level and were not under a subdirectory were not supported and did not work with HCL OneTest™ Server. This problem is fixed.

Defects fixed in 10.1.0

The defects fixed in HCL OneTest™ Server 10.1.0 are as follows:

Table 15. List of fixes in 10.1.0

ID	Description
TP-66237	Previously, HCL OneTest™ Server projects were sorted by date. Now, server projects are sorted alphabetically.

Table 15. List of fixes in 10.1.0 (continued)

ID	Description
TP-63775	<p>Previously, in HCL OneTest™ Server, it was not clear that classifications could only be made for encrypted datasets.</p> <p>This problem is fixed.</p>
TP-63156	<p>Previously, Docker images that were created with no tag were assigned the <i>latest</i> tag. As the docker ps command did not show the <i>latest</i> tag anymore since the Docker 19.03.05 release, the Docker images were not detected and were not correctly monitored.</p> <p>This problem is fixed.</p>
TP-63149	<p>Previously, intermediate <i><none>:<none></i> Docker images were represented as is and could not be selected individually in Resource Monitoring counters.</p> <p>This problem is fixed.</p>
TP-62643	<p>If you backed up a project with both API suites and stubs on HCL OneTest™ Server 10.0.0, when you restored that backup on HCL OneTest™ Server 10.0.2, the stubs were missing.</p> <p>This problem is fixed.</p>
TP-62571	<p>If you wanted to add multiple repositories to a project in HCL OneTest™ Server you could not add another repository until the last repository completed its synchronization.</p> <p>This problem is fixed.</p>
TP-62417	<p>Previously, a user who was no longer a member of a project might be displayed in the Secrets tab. If a user was removed from a project and then added back to that project, the user would have the same access to secrets as they had before being removed.</p> <p>This problem is fixed.</p>
TP-62338	<p>Previously, an IBM WebSphere MQ stub that was configured to use more than one transport did not appear on the Execution page of the HCL OneTest™ Server.</p> <p>This problem is fixed.</p>
TP-61897	<p>Previously, if you restored a backup, it failed due to existing execution containers.</p>

Table 15. List of fixes in 10.1.0 (continued)

ID	Description
	This problem is fixed.
TP-61786	<p>Previously, when you entered a URL for a test result directly into a browser or you clicked a link to a test result in the Jenkins console output from the Jenkins HCL OneTest™ Server plugin, and you were not logged in to HCL OneTest™ Server, you saw the following message:</p> <p>Request cannot be processed: Unauthorized page.</p> <p>This problem is fixed.</p>
TP-61179	<p>Previously, executing a test on a non-English system failed.</p> <p>This problem is fixed.</p>
TP-61153	<p>Previously, the project search field was case-sensitive. For example, if your project was named <i>MYPROJECT</i> and you searched for <i>myproject</i>, HCL OneTest™ Server did not show any results.</p> <p>This problem is fixed.</p>
TP-58125	<p>Previously, HCL OneTest™ Server did not release system resources after they had exited.</p> <p>This problem is fixed.</p>
TP-57543	<p>Previously, if the server hostname was changed, it was necessary to run <code>./prepare.sh</code> and log in to the Keycloak Admin Console to change the testserver client URLs. Otherwise the message <code>Invalid parameter: redirect_uri</code> was displayed on the Login page.</p> <p>This problem is fixed.</p>

Known issues

You can find the known issues that are identified, if any, in this version of HCL DevOps Test Hub (Test Hub). You can also find the known issues identified in the earlier versions of Test Hub. DevOps Test Hub is the new name of HCL OneTest™ Server from 11.0.0 onwards.

You can find the following information:

- [Known issues in DevOps Test Hub 2024.03 \(11.0.1\) on page 53](#)
- [Known issues from earlier versions on page 53](#)

Known issues in DevOps Test Hub 2024.03 (11.0.1)

The known issues identified in Test Hub 2024.03 (11.0.1) are as follows:

Table 16. Known issues in Test Hub 2024.03 (11.0.1)

ID	Description
TP-92219	<p>The test log for a JMeter test might be unavailable if the default trust store of the JVM is altered by the test.</p> <p>To work around this problem, you must perform the following steps when you run each JMeter test:</p> <ol style="list-style-type: none"> 1. Select the JMeter test, and then click Execute. <p style="padding-left: 40px;">The Execute test asset dialog is displayed.</p> <ol style="list-style-type: none"> 2. Click Advanced. 3. Enter <code>CISTERNA_CONFIG_APPEND</code> in the first column, and <code>&trustStorePath=/opt/java/openjdk/lib/security/cacerts</code> in the second column in the Additional Configuration Parameters field. 4. Click Add. 5. Click Execute. <p style="padding-left: 40px;">A message is displayed that confirms the initiation of the test execution.</p>

Known issues from earlier versions

The known issues identified in the earlier versions of Test Hub that are still applicable are as follows:

Table 17. Known issues from earlier versions

ID	Description	Identified in version	Applicable in and until version
TP-90652	You cannot run a Windows test on agent computers that are selected as the location to run the test when you execute the test in Test Hub 11.0.0.	11.0.0	
TP-89553	The unified report is displayed incorrectly when you run a Compound Test that contains mobile tests to run on either the devices, emulators, or simulators that are con-	10.5.4	10.5.4

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	<p>ected to an agent computer or the devices that are connected to a mobile device cloud.</p> <p>To enable the unified report to display correctly, you must provide the program argument -history testlog in the Advanced configuration when you configure a run of such Compound Tests.</p>		
TP-75715	HCL OneTest™ Server does not support the sorting of the test results on the Results page based on the person who started the test run when you click the Started by column header.	10.5.4	10.5.4
TP-88328	<p>The data generation does not generate test data when JavaScript rules are used to create item types.</p> <p>Also, the data generation feature does not validate the Regular expressions used to define item values in a schema.</p>	10.5.3	10.5.3
TP-88027	You cannot perform a direct upgrade of HCL OneTest™ Server 10.5.3 from versions 10.5.1 or earlier.	10.5.3	10.5.3
TP-88111	The default configuration in Keycloak incorrectly assigns all users of HCL OneTest™ Server the privileges of a server administrator.	10.5.2	10.5.2
TP-88079	Members of a team space are able to view the details of a project when they enter the URL of a page in a project in spite of not being a member of that project.	10.5.2	10.5.2
TP-86722	HCL OneTest™ Server incorrectly creates duplicate records of the test runs in the integrated Xray-Jira project.	10.5.2	10.5.2

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
TP-86633	You cannot perform a direct upgrade of HCL OneTest™ Server 10.5.2 from versions 10.2.2 or earlier.	10.5.2	10.5.2
TP-86087	When programmatically starting a large number of test assets at once, users should be mindful of the capacity of the system, size of projects and adjust the rate accordingly.	10.5.2	10.5.4
TP-85868	Virtualization usage graphs at the team space or project level in HCL OneTest™ Server 10.5.2 are not displayed for virtual services that ran on an earlier version of the server.	10.5.2	
TP-81529	When you run and complete a test by using the HCL OneTest™ Server UCD plugin, the UCD log displays the state of the test with a lag in real time. For example, the state is displayed as "In transition" when the test is in the "Running" state.	10.5.2	
TP-61446	<p>When you unregister a remote docker agent, running and schedule tests are not canceled.</p> <p>To workaround this problem, stop any running tests and cancel any scheduled future executions for that agent before unregistering the remote docker agent.</p>	10.5.2	
TP-85354	<p>When you perform bulk update of test assets by using the test authoring feature on HCL OneTest™ Server, you encounter timeout errors when you attempt to commit or publish the changed assets.</p> <p>To work around this problem, you can commit or publish the assets in smaller batches.</p>	10.5.1	10.5.4
TP-78142	You cannot migrate the data about the IBM Engineering Test Management that is configured as a quality man-	10.5.1	10.5.1

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	agement solution on HCL OneTest™ Server 10.5.0 to HCL OneTest™ Server 10.5.1.		
TP-65557	When you created projects in HCL OneTest™ Server 10.5.0 or earlier by using LDAP/AD, but did not import the LDAP/AD users into the migrated projects in 10.5.1, and deleted the users in the project, the project cannot be opened or assigned to other users.	10.5.1	10.5.1
TP-84473	<p>When you configure IBM® Engineering Test Management and Atlassian Jira on HCL OneTest™ Server 10.2.3, the information about Engineering Test Management and Jira is not visible after you migrate HCL OneTest™ Server from 10.2.3 to 10.5.0.</p> <p>To work around this problem, you must enter the details of Engineering Test Management and Jira again on HCL OneTest™ Server 10.5.0.</p>	10.5.0	10.5.0
TP-84027	<p>When you run a schedule that has a custom code to write data into a dataset from a non-Edit branch of HCL OneTest™ Server, the Changes page does not list the updated dataset to publish the changes made to the dataset.</p> <p>To work around this problem, you must run the schedule from an Edit branch.</p>	10.5.0	
TP-83997	When you run a compound test that contains an API test on HCL OneTest™ Server and the API test uses a dataset or a different API test is selected at the time of test configuration, the unified report for the compound test does not provide the correct or complete information about the API test.	10.5.0	
TP-80168	When you create a schema, the screen reader incorrectly calls the 'Expand' action as 'Maximize' and 'Collapse' ac-	10.5.0	10.5.3

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	tion as 'Minimize' in the 'Dictionary', 'Structure', and 'Properties' panels.		
TP-80166	The icons for 'Item' type and 'Group' type objects in a schema fail to meet the AA and AAA contrast (between foreground and background) requirements.	10.5.0	10.5.3
TP-78225	You cannot migrate edited test assets that were created in earlier versions to HCL OneTest™ Server 10.5.	10.5.0	
TP-78170	The generation of test data with a large number of records into an spreadsheet in the .xls format sometimes fails.	10.5.0	10.5.3
TP-77323	HCL® OneTest™ Data fails to generate data in the XSD format when you provide specific seed values.	10.5.0	10.5.3
TP-75894	The 'Root' element of a schema in Data Fabrication cannot be accessed through keyboard actions.	10.5.0	10.5.3
TP-61311	When you archive a project that has a running test, that test is not stopped. The test completes execution even though it was suppose to stop.	10.5.0	
TP-82448	When you run a performance test that contains integration tests on HCL OneTest™ Server, the test log captures and displays all integration test steps under the first activity that was captured for the performance test.	10.2.3	10.2.3
TP-81800	There is a time delay for the changes that you make to your personal details provided at the time of signing up on HCL OneTest™ Server to be reflected in the UI.	10.2.3	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
TP-81017	The editing session for an API Suite is not launched when you attempt to edit the asset that was committed to remote repository with non-omitted spaces in its path.	10.2.2	10.2.2
TP-80637	<p>While you install HCL OneTest™ Server on Azure Kubernetes Service (AKS), if the postgresql-init job does not run early, then the postgresql-init job remains in a Pending state due to insufficient CPU resources.</p> <p>To work around this problem, you must delete the pods controlled by deployments that make the required resources temporarily available.</p>	10.2.2	10.2.2
TP-80565	<p>When you deploy HCL OneTest™ Server on Azure Kubernetes Service (AKS), and if you do not set up the Domain Name Server (DNS) resolution of the client, then you cannot open a new editor session in HCL OneTest™ Server. The browser displays an error message that the IP address of the editor hosts is not found.</p> <p>To work around this problem, you must map the host-name details of the editor that are displayed in the error message with the server IP address in the hosts file. You can locate the hosts file in the following location on Windows systems: <code>C:\Windows\System32\drivers\etc</code></p>	10.2.2	10.2.2
TP-80471	HCL OneTest™ Server supports any Git repository accessible over the HTTP or SSH protocols. When you add or reconfigure a repository that is associated with a project, HCL OneTest™ Server verifies the configuration by attempting to connect and issuing a request to the given URI of the repository. You must ensure that only trusted URIs are provided as part of the configuration.	10.2.2	10.2.2
TP-80470	The default configuration of the HCL OneTest™ Server provides access to requests that are coming from any	10.2.2	10.2.2

Table 17. Known issues from earlier versions (continued)


ID	Description	Identified in version	Applicable in and until version
	<p>domain. The request can be integrated into other web servers, and exposes HCL OneTest™ Server to Cross-origin resource sharing (CORS) attacks.</p> <p>To work around this problem, you must overwrite the configuration properties value during the installation of the HCL OneTest™ Server to restrict CORS to a specific allowlist by adding the following helm argument:--set-string 'gateway.extraEnv.JAVA_TOOL_OPTIONS=Dgateway.cors.allowed-origin-patterns=*.example.com,*mydomain.net'Where example.com and mydomain.net are the domains that you want to provide access to CORS.</p> <p> Note: You can provide multiple domain names separated by comma (,).</p>		
TP-78678	<p>You cannot deploy HCL OneTest™ Server 10.2.1 on Azure Kubernetes Service (AKS) if you have installed the latest version of the NGINX Ingress Controller.</p> <p>To work around this problem, you must install the NGINX Ingress Controller by using the helm chart 3.26.0.</p>	10.2.1	10.2.1
TP-78225	<p>If you edited test assets in your project on HCL OneTest™ Server 10.2 and then migrated to HCL OneTest™ Server 10.2.1. After migration if there were changes added to the remote repository related to the edited test assets then the values displayed in the Last Changed By field might be incorrect.</p>	10.2.1	10.2.3
TP-78219	<p>If you update the name of the project in Jira that you used to configure on HCL OneTest™ Server, you cannot configure the updated project name in the CHANGE MANAGEMENT SYSTEM tab of HCL OneTest™ Server.</p>	10.2.1	10.2.1
TP-78146	<p>If you update or delete a Project area that you used to configure Engineering Test Management on HCL</p>	10.2.1	10.5.0

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	<p>OneTest™ Server, then you cannot perform the following tasks:</p> <ul style="list-style-type: none"> • Delete your project from HCL OneTest™ Server. • Delete the configured Engineering Test Management from the HCL OneTest™ Server project. 		
TP-78028	The Engineering Test Management displays the status as Connected on HCL OneTest™ Server even when Engineering Test Management is unavailable for any reason.	10.2.1	
TP-76628	<p>In the Azure environment, when you generate the test data for a schema with a regular expression property set for an item type, then the job to generate the test data fails.</p> <p>To work around this problem, you must use a sample file that contains values for the restriction instead of setting the regular expression property for any item type.</p>	10.2.0	10.5.3
TP-76496	<p>When you restart the pods, the dataset service might fail to restart due to a lock on the dataset tables.</p> <p>To work around this problem, you must run the following commands before you restart the Kubernetes cluster:</p> <ol style="list-style-type: none"> 1. Run the following command to get the password for the dataset service: <pre>kubectl get secret (my-server)-datasets -o yaml -n {namespace}</pre> 2. Run the following command to remove the lock from the dataset table: <pre>kubectl exec (my-server)-postgresql-0 -n {namespace} -i - sh -c 'PGPASSWORD=<password> psql -U datasets -d datasets -c "DROP TABLE databasechangelock"'</pre> 	10.2.0	10.2.0

Table 17. Known issues from earlier versions (continued)


ID	Description	Identified in version	Applicable in and until version
	<p>3. Run the following command to restart the Kubernetes cluster:</p> <pre>sudo systemctl start k3s</pre> <p> Note: You must substitute the following variables with the actual value in the commands:</p> <ul style="list-style-type: none"> ◦ <i>{my-server}</i> with the release name of the server that you provided during the installation of the server software. ◦ <i>{namespace}</i> with the name of the namespace that you created during the installation of Test Hub. ◦ <i><password></i> with the value that you obtained from Step 1. 		
TP-76105	<p>TLS1.1 and TLS1 are not supported on HCL OneTest™ Server. When you try to run those test assets, the test run fails with an error message about an unsupported security algorithm.</p>	10.2.0	10.2.0
TP-74890	<p>After you upgrade HCL OneTest™ Server from V10.1.2 to V10.1.3, then you might fail to generate test data.</p> <p>To work around this problem, you must perform the following steps:</p> <ol style="list-style-type: none"> 1. Get the persistence volume claims (pvc) with namespace: <code>kubectrl get pvc -n <my-ots></code> 2. Describe the pvc to get the path of the volume: <code>kubectrl describe pv <volume></code> 3. Provide access permissions to the volume claim: <code>sudo chmod 777 -R <path></code> 4. Change the ownership of the HIP volume: <code>sudo chown -R root:root <path></code> 	10.1.3	10.5.3

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	<p>5. Describe the pvc to get the path of the volume: kubectrl describe pv <volume number></p> <p>6. Provide access permissions to the volume claim: sudo chmod 777 -R <path></p> <p>7. Change the ownership of the data app volume: sudo chown -R root:root <path></p> <p>8. Run the following command: kubectrl get pods -n <test-system></p> <p>9. Restart the pods by using the following command: kubectrl delete pods <my-ots-hip-rest> my-ots-data-app-0 my-ots-hip-server-redis-master-0 my-ots-hip-server-0 -n <test-system></p> <p>10. Run the following commands to view the pods:</p> <p>kubectrl get pods -n <test-system></p> <p>watch kubectrl get pods -n <test-system></p>		
TP-74534	<p>Before you migrate user data from an earlier version to the latest version, you must publish any created datasets to the Git repository. After the migration, when you try to publish a created dataset it might fail. If it does fail, your only option is to recreate the dataset from scratch.</p>	10.1.3	10.2.0
TP-74329	<p>When you upgrade HCL OneTest™ Server from V10.1.1 to V10.1.3, then you might observe issues in the generation of test data on the migrated projects.</p> <p>To work around this problem, you must export all your HCL® OneTest™ Data projects from V10.1.1 to your computer, and then import the projects to V10.1.3.</p>	10.1.3	10.1.3
TP-74116	<p>After you import any JSON schema into the HCL® OneTest™ Data project, and if you modify the basic properties of the item types, then you might face issues to generate test data.</p>	10.1.3	10.5.3

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
TP-73109	When you install HCL OneTest™ Server on Lightweight Kubernetes (k3s) and warnings about Ingress deprecation are displayed, you can ignore these warnings.	10.1.3	10.2.0
TP-71416	When you attempt to run an API Suite on HCL OneTest™ Server that is installed on OpenShift, and when the API project contains a vast set of resources, you might see that the Execute button in the Execute test asset dialog box is enabled after a delay or at times is not enabled at all.	10.1.3	10.1.3
TP-72106	When you expand a dataset from the Datasets page, you cannot navigate to all the cells in the dataset by using the keyboard keys.	10.1.2	10.5.3
TP-71956	<p>When you have HCL OneTest™ Server on Ubuntu and you run performance test assets such as VU schedule or Rate schedule on a static agent, then the test runs fail with an error message.</p> <p>To work around this problem, you must add the following parameter to the helm install command during the installation of HCL OneTest™ Server:</p> <pre>--set networkPolicy.enabled=false</pre> <p>If you have already installed HCL OneTest™ Server, you can modify the helm parameter value by using the following commands:</p> <pre>helm upgrade {{main}} ./ ibm-rtas-prod -n test-system \ --reuse-values --set networkPolicy.enabled=false</pre> <p>Where <i>main</i> is the name of the release that you provided during the installation of HCL OneTest™ Server.</p>	10.1.2	10.1.2

Table 17. Known issues from earlier versions (continued)


ID	Description	Identified in version	Applicable in and until version
TP-71950	<p>When the server is under high load, the results microservice might be restarted indefinitely. This happens when the server was killed while holding a database lock.</p> <p>To work around the problem, perform the following steps:</p> <p> Note: In the following commands, <myrtas> refers to the release name of your choice and <namespace> refers to the name of the namespace that you created during the server installation process.</p> <ol style="list-style-type: none"> 1. Increase the value of failureThreshold in the following command to start the service: <code>kubectl patch sts -n <namespace> <myrtas>-results --patch '{"spec": {"template": {"spec": {"containers": {"name": "results", "startupProbe": {"failureThreshold": 120}}}}}'</code> 2. Get the database password by running the following command: <code>kubectl get secret -n <namespace> <myrtas>-results -o jsonpath="{.data.postgresql-password}" base64 --decode</code> 3. Delete the troublesome database lock by running the following commands: <code>kubectl exec -n <namespace> <myrtas>-results-postgresql-0 -i - sh -c 'PGPASSWORD=<PASSWORD> psql -U results -d results -c "DROP TABLE DATABASECHANGELOGLOCK"'</code> <code>kubectl exec -n <namespace> <myrtas>-results-postgresql-0 -i - sh -c 'PGPASSWORD=<PASSWORD> psql -U results -d analytics -c "DROP TABLE DATABASECHANGELOGLOCK"'</code> <code>kubectl exec -n <namespace> <myrtas>-results-postgresql-0 -i - sh -c 'PGPASS-</code> 	10.1.2	10.1.3

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	<p>WORD=<PASSWORD> psql -U results -d upload -c "DROP TABLE DATABASECHANGELOGLOCK"</p> <p>4. Restart the pod by running the following command: kubectl delete pod -n <namespace> <myr-tas>-results-0</p>		
TP-71416	<p>When you attempt to run an API Suite on HCL OneTest™ Server that is installed on OpenShift, and when the API project contains a vast set of resources, you might see that the Execute button in the Execute test asset dialog box is enabled after a delay.</p>	10.1.2	10.1.3
TP-71030	<p>After you install the product, if pods restart with the following log message:</p> <pre data-bbox="418 940 1003 1010">ACCESS_REFUSED - Login was refused using the authentication mechanism PLAIN</pre> <p>For details see the broker logfile. You must uninstall the product, delete the pvc data-{{main}}-rabbitmq-0, and then install the product.</p>	10.1.2	10.1.2
TP-71004	<p>Stubs are a tech-preview feature, as such some functionality might not work in all environments.</p>	10.1.2	10.2.3
TP-69141	<p>When you install V10.1.2 of the product, you must decrypt and encrypt each dataset by using the CSV Editor to apply the strongest encryption on each dataset that you used in the earlier version of the product.</p>	10.1.2	10.5.3
TP-67572	<p>If a dataset is created on the server using a path within an integration tester project and the created CSV file is subsequently referenced in a file data source, then the server presents two datasets relating to the same CSV file.</p>	10.1.2	10.5.3
TP-70414	<p>When you install HCL OneTest™ Server V10.1.1 on an OpenShift platform where OpenShift Container Storage</p>	10.1.1	10.1.1

Table 17. Known issues from earlier versions (continued)


ID	Description	Identified in version	Applicable in and until version
	<p>(OCS) is not available (For example, the IBM Cloud) then test asset runs fail. This is because the default storage class does not support ReadWriteMany (RWX).</p> <p>To work around this problem, you must specify the following helm parameter during the server install:</p> <pre data-bbox="418 590 1003 680">--set execution.userlibs.persistence.storageClass=ib mc-file-bronze</pre>		
TP-69650	<p>When you run the ubuntu-init.sh script to set up the Kubernetes environment and if the snap command is unavailable, then the command-line interface incorrectly displays the following error message for snapd: <code>./ubuntu-init.sh: line 232: ehco: command not found</code> The intended message is: <code>Error: snapd is not installed</code></p> <p>Please ensure that snapd and snap are installed correctly, for example: <code>sudo apt-get install snapd</code></p>	10.1.1	
TP-69424	<p>When you back up user data of V10.1.0 and restore it on V10.1.1 in the Ubuntu environment, one of the pods does not start. This problem occurs occasionally.</p> <p>For example, if the result pod does not start, to work around this problem, perform the following steps:</p> <p> Note: In the following commands, <code><myrtas></code> refers to the release name of your choice and <code><namespace></code> refers to the name of the namespace that you created during the server installation process.</p> <ol style="list-style-type: none"> 1. Increase the value of <code>initialDelaySeconds</code> in the following command to start the service: <pre data-bbox="483 1755 992 1820">kubectl patch sts -n <namespace> <myrtas>-results --patch '{"spec": {"template": {"spec": {"con-</pre> 	10.1.1	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	<p>ainers": [{"name": "results", "readinessProbe": {"initialDelaySeconds": 180}}}]}'</p> <p>2. Get the database password by running the following command:</p> <pre>kubectl get secret -n <namespace> <myrtas>-results -o jsonpath="{.data.postgresql-password}" base64 --decode</pre> <p>3. Delete the troublesome database lock by running the following command:</p> <pre>kubectl exec -n <namespace> <myrtas>-results-postgresql-0 -i -- sh -c 'PGPASSWORD=<password> psql -U results -d results -c "DROP TABLE DATABASECHANGELOGLOCK"'</pre> <p>4. Restart the pod by running the following command:</p> <pre>kubectl delete pod -n <namespace> <myrtas>-results-0</pre>		
TP-66941	You cannot run tests on the remote static agents connected to HCL OneTest™ Server that is installed in an OpenShift environment.	10.1.1	10.1.1
TP-66834	<p>During the installation, when you migrate user data from the previous version to V10.1.0, the datasets database does not restore properly. As a result, Data classification information disappears from HCL OneTest™ Server.</p> <p>You must classify the encrypted dataset again by creating a new classification.</p>	10.1.1	10.1.1
TP-66731	You might experience errors when you publish large reports, either from desktop products or when executing a test on the server.	10.1.1	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	<p>To circumvent this problem, it is recommended that you first try to reduce your reports size by reducing the logging or details level. If the detail level cannot be reduced, you can increase the server limits to support larger reports. To do so, you must rerun the helm command that was performed when you installed Test Hub and add the following arguments:</p> <pre data-bbox="418 646 1003 852"> --set-string ingress.annotations.nginx\\.ingress\\ \\.kubernetes\\.io/proxy-body-size=2g --set extraEnv.SPRING_SERVLET_MULTIPART_MAX_REQUEST_SIZE=2GB --set resources.limits.memory=4Gi </pre> <p>This example enables the server to accept up to 2 GB reports. This comes at the expense of a higher memory requirement. It is usually recommended to set the resources.limits.memory to 2 GB over the reports size limit.</p>		
TP-67215	<p>When you configure HCL OneTest™ Server test on the UCD server and you enter a value for any of the options available for Show Hidden Properties in the Edit Properties for Run HCL OneTest™ Server test window, the values for the options do not work as designed during the test run.</p> <p>To work around this problem, you must not set any values for any options for Show Hidden Properties.</p>	10.1.0	
TP-66941	You cannot run tests on the remote static agents connected to HCL OneTest™ Server that is installed in an OpenShift environment.	10.1.0	
TP-64528	When you ran either a Rate or a VU Schedule on HCL OneTest™ Server and selected a disconnected agent or multiple agents to run on, the run failed and displayed an	10.1.0	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	Error 500 but the execution job stayed in the <i>In Transition</i> state.		
TP-61494	<p>If you unarchive a project that was archived while a test was running, the UI incorrectly shows that the test is still running after the unarchive.</p> <p>To workaround this problem, make sure there are no tests running when you archive a project.</p>	10.1.0	
TP-59943	<p>If you configure Keycloak to use an LDAP directory service, you must import and synchronize LDAP users to Keycloak. This is a known issue with Keycloak: https://issues.redhat.com/browse/KEYCLOAK-13822</p> <p>To workaround the problem in Keycloak, on the Add user federation provider page make sure the following settings are set as shown: Import Users = ON, Sync Registrations = ON, and Cache Policy = DEFAULT</p>	10.1.0	
TP-63794	When you back up the server, large amounts of additional disk space can be used. To reduce the disk space used, you should first remove the execution container volumes.	10.0.2.1	
TP-63156	Docker images that are created with no tag are assigned the "latest" tag. As the "docker ps" command does not show the "latest" tag anymore since the Docker 19.03.05 release, the Docker images are not detected and not correctly monitored.	10.0.2.1	
TP-63149	Intermediate <none>:<none> Docker images are represented as is and cannot be selected individually in Resource Monitoring counters.	10.0.2.1	
TP-62729	When upgrading from a previous release to V10.0.2, users might see a 500 error when accessing datasets.	10.0.2	10.0.2

Table 17. Known issues from earlier versions (continued)


ID	Description	Identified in version	Applicable in and until version
	<p>This problem can be resolved by performing the following step: Add the following environment variable to automation/docker-compose.yml in the HCL OneTest™ Server installation directory. The line must go under the "datasets-app: environment:" section (located around line 128). Then reboot the server. <code>SPRING_JPA_HIBERNATE_DDL-AUTO: create</code></p> <p> Note: Adding this environment variable will cause saved datasets options (for example, current row) to be reset to the default upon server startup. Other migrated data will be unaffected. The environment variable must be removed once the issue is resolved.</p>		
TP-62643	<p>If you back up a project with both API suites and stubs on HCL OneTest™ Server V10.0.0, when you restore that backup on HCL OneTest™ Server V10.0.2, the stubs are missing.</p> <p>To workaround this problem, you can create a new project or you can move the existing GIT repository folder to a sub folder of the current repository and then move it back to the root directory. For example, a project contains a Git repository called Calculator.git in the root (/) folder. When you edit the root folder, you can see several other Git repository folders: <code>/GitNewProjectMyDemo-Calculator</code></p> <p>You select another folder, for example Calculator, and save. The Git repository called Calculator.git is now in the <code>Calculator</code> folder. You edit the Calculator folder, select the root folder, and save. The Git repository is now in the root (/) folder again.</p>	10.0.2	
TP-62571	If you want to add multiple repositories to a HCL OneTest™ Server project, you cannot add another repos-	10.0.2	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	<p>itory until the last repository completes its synchronization.</p> <p>To workaround this problem, wait for the last repository to synchronize, then add another repository to your project.</p>		
TP-62417	<p>A user who is no longer a member of a project might be displayed in the Secrets tab. If a user is removed from a project and then added back to that project, the user will have the same access to secrets as they had before being removed.</p>	10.0.2	
TP-62379	<p>If you (as a user) request to join an orphaned project, a 404 not found error results.</p> <p>To workaround this problem, ask your administrator to become the project owner, then request to join that project again.</p>	10.0.2	
TP-62338	<p>An IBM WebSphere MQ stub that is configured to use more than one transport does not appear on the Execution page of the HCL OneTest™ Server.</p>	10.0.2	
TP-61897	<p>If you restore a backup, it fails due to existing execution containers.</p>	10.0.2	
TP-61786	<p>When you enter a URL for a test result directly into a browser or you click a link to a test result in the Jenkins console output from the Jenkins HCL OneTest™ Server plugin, and you are not logged in to HCL OneTest™ Server, you see the following message: <code>Request cannot be processed: Unauthorized page.</code></p> <p>To workaround this problem, log in to HCL OneTest™ Server, and then enter the URL or click the test result link to see the test result in a browser.</p>	10.0.2	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
TP-61494	<p>If you unarchive a project that was archived while a test was running, the UI incorrectly shows that the test is still running after the unarchive.</p> <p>To workaround this problem, make sure there are no tests running when you archive a project.</p>	10.0.2	
TP-61446	<p>When you unregister a remote docker agent, running and schedule tests are not canceled.</p> <p>To workaround this problem, stop any running tests and cancel any scheduled future executions for that agent before unregistering the remote docker agent.</p>	10.0.2	
TP-61311	<p>When you archive a project that has a running test, that test is not stopped. The test completes execution even though it was suppose to stop.</p>	10.0.2	
TP-61153	<p>The project search field is case-sensitive. For example, if your project is named MYPROJECT and you search for myproject, HCL OneTest™ Server does not show any results.</p>	10.0.2	
TP-60737	<p>Git repositories that contain projects at the root level and not under a subdirectory are not supported and will not work currently with HCL OneTest™ Server.</p>	10.0.2	
TP-60059	<p>When you schedule a test run and click the Time field a list of times displays. If you enter a value in the Time field, you cannot move the cursor by using the left or right (<- , ->) keyboard keys. This is a known issue with the following date picker:https://github.com/Hacker0x01/react-datepicker/issues/1834</p> <p>To workaround this problem, use the cursor.</p>	10.0.2	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
TP-60059	<p>When you schedule a test run and click the "Time" field a list of times displays. If you enter a value in the "Time" field, you cannot move the cursor by using the left or right (<, >) keyboard keys. This is a known issue with the following date picker:https://github.com/Hacker0x01/react-datepicker/issues/1834</p> <p>To workaround this problem, use the cursor.</p>	10.0.1	
TP-59943	<p>If you configure Keycloak to use an LDAP directory service, you should import and synchronize LDAP users to Keycloak. This is a known issue with Keycloak.</p> <p>To workaround the problem in Keycloak, on the Add user federation provider page make sure the following settings are set as shown: Import Users = ON, Sync Registrations = ON, and Cache Policy = DEFAULT</p>	10.0.1	
TP-59774	<p>When checking in a project without a root directory an error is thrown while at the Execution dialog box that does not allow you to select an environment, nor does it remember your selection of secrets. Do not check-in desktop projects to the root of the Git repository. All checked-in desktop projects must contain a parent directory within the Git repository. For example, if the desktop project contained the following assets: <code>MyProject/.project-MyProject/MyTest.testsuite</code></p> <p>Then the Git repository should contain <code>MyProject/.project-MyProject/MyTest.testsuite</code> and not just <code>MyProject/.project</code> -> Will cause an error/MyTest.testsuite</p>	10.0.1	
TP-59139	<p>During the installation of HCL OneTest™ Server the following error message is shown and can be safely ignored: <code>139836115718592:error:2406F079:random number</code></p>	10.0.1	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:88:Filename=/root/.rnd		
TP-58572	Execution results with status "Complete with Error" or "Launch failed" are displayed with inconsistent colors. In verdict statistics views (in Overview and Results pages) they are displayed in blue, whereas in the detail tables they are displayed in red.	10.0.1	
TP-58017	In case of a slow network, the cloning of a repository might fail with an error. In such a case, you can delete and add the repository again.	10.0.1	
TP-58292	If you interrupt the installation or uninstallation of the HCL OneTest™ Server software, the installation or uninstallation cannot complete. For a workaround, see Troubleshooting installation.	10.0.0	
TP-58170	Selecting a date in the Execution Results card on the Project Overview page does not update the results for that date.	10.0.0	
TP-58125	<p>HCL OneTest™ Server does not currently remove Docker containers after they have exited. The containers will exist in a 'stopped' state but should consume only disk resources (not memory). The Docker command <code>docker ps -a</code> shows both running and stopped containers.</p> <p>This problem is normally not an issue but if the list grows large it consumes disk resources or might make it difficult to find other stopped containers.</p> <p>On HCL OneTest™ Server system (assuming Linux), to view stopped execution containers, use the following Docker command:</p> <pre>docker ps --filter "status=exited" grep 'EC_[0-9]' awk '</pre>	10.0.0	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	<pre>{print \$1}' To remove stopped execution containers from the system, use the following Docker command: docker ps --filter "status=exited" grep 'EC_[0-9]' awk '{print \$1}' xargs docker rm</pre>		
TP-58104	When monitoring a test execution with the Statistical report, data might stop updating after about an hour. To get updated data, reload the page in your browser.	10.0.0	
TP-58061	A server shutdown will not stop currently running executions. It is recommended to not shutdown the server if there are active executions running. If the server is shut down while assets are executing the reported status of those executions might be incorrect.	10.0.0	
TP-58061	A server shutdown will not stop currently running executions. It is recommended to not shutdown the server if there are active executions running. If the server is shut down while assets are executing the reported status of those executions might be incorrect.	10.0.0	10.1.2
TP-58024	HCL OneTest™ Server Login and Account Settings pages are only available in English.	10.0.0	
TP-58017	In case of a slow network, the cloning of a repository might fail with an error. In such a case, you can delete and add the repository again.	10.0.0	
TP-58017	In case of a slow network, the cloning of a repository might fail with an error. In such a case, you can delete and add the repository again.	10.0.0	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
TP-58016	<p>If there are errors encountered during server shutdown or while uninstalling the server, they can be caused by active asset executions in progress. It is recommended to allow any executions in progress to complete before shutting down or uninstalling the server. If you want to force a shutdown, the following commands can be used to identify and stop active containers:</p> <p>Show all running and stopped containers that are part of server asset execution:</p> <pre>docker ps -a -f 'name=EC_*</pre> <p>Remove containers, including running containers:</p> <pre>docker rm -f \$(docker ps -q -a -f 'name=EC_*</pre>	10.0.0	
TP-57971	<p>After adding a repository to a project, you see a GREEN status indicator. When you remove the ssh key from Github, the GREEN status changes to RED. When you update Github with a new ssh key and update the same for the repository in HCL OneTest™ Server, the status indicator does not immediately change to GREEN but it does change to GREEN after a couple of minutes.</p>	10.0.0	
TP-57832	<p>HCL OneTest™ Server for Windows is provided for evaluation purposes. It is intended for non-production use, for example, trials and demonstrations. It is neither supported nor recommended for production use due to insufficient stability and reliability of Docker for Windows while running Linux images at the time of release.</p> <p>HCL OneTest™ Server is tested on both Windows 10 and Windows Server 2016 Standard Edition and it operates correctly. However, it is suitable and supported for non-production use only. If native Linux is unavailable, you can run HCL OneTest™ Server for production use on a Linux virtual machine on their Windows host, provided</p>	10.0.0	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	that the Linux virtual machine is running a supported version of Linux.		
TP-57198	<p>When working in multiple HCL OneTest™ Server tabs concurrently, a message “You are already logged in” might sometimes be displayed. This message occurs when your session has expired and you have since logged in again, but then attempt a navigation action in a browser tab other than the one in which you logged in.</p> <p>This is a known Keycloak issue. For more information, see https://issues.jboss.org/browse/KEYCLOAK-5179. To workaround this problem, close the tab and open a new one.</p>	10.0.0	
TP-57198	<p>When working in multiple HCL OneTest™ Server tabs concurrently, a message <code>You are already logged in</code> might sometimes be displayed. This message occurs when your session has expired and you have since logged in again, but then attempt a navigation action in a browser tab other than the one in which you logged in.</p> <p>This is a known Keycloak issue. For more information, see https://issues.jboss.org/browse/KEYCLOAK-5179. To workaround this problem, close the tab and open a new one.</p>	10.0.0	
TP-56453	After about 20 minutes of an inactive browser session for HCL OneTest™ Server, if you attempt to use the existing browser session, you must authenticate again by providing your username and password. When you do so, the message “You took too long to log in. Try again.” is displayed and you must enter your credentials again.	10.0.0	
HCL OneTest™ Performance Agent	With 10.0, you might expect a reduction in the number of Virtual Users (VUs) supported by HCL OneTest™ Performance Agent depending upon workload characteristics.	10.0.0	

Table 17. Known issues from earlier versions (continued)

ID	Description	Identified in version	Applicable in and until version
	<p>This variation is the result of the product shipping with OpenJDK™ 8 with OpenJ9 instead of IBM JRE.</p> <p>OpenJDK 8 with OpenJ9 JRE exhibits a higher utilization of CPU for the workload. Up to 30% or more decrease in the supported VUs capacity may be expected for CPU-bound workloads. However, for memory-bound workloads, a much smaller, if any reduction in VUs capacity may be expected.</p> <p>If you need higher performance and have access to IBM JRE 8.0.5.35 or later that contains the latest security fixes, you can switch from OpenJDK 8 with OpenJ9 to IBM JRE. For more information, see http://www.ibm.com/support/docview.wss?uid=ibm10956413.</p>		

Chapter 2. System Requirements for DevOps Test Hub 2024.03 (11.0.1)

This document includes information about hardware and software requirements for HCL DevOps Test Hub (Test Hub).

Contents

- [Hardware on page 80](#)
- [Operating systems and Containers on page 80](#)
 - [Bit version support on page 81](#)
 - [Operating systems on page 81](#)
 - [Containers on page 81](#)
- [Host prerequisites on page 82](#)
 - [Licensing on page 82](#)
- [Supported software on page 82](#)
 - [Development tools on page 83](#)
 - [Integrations on page 83](#)
 - [Runtime environment on page 85](#)
 - [Web browsers on page 85](#)
- [Disclaimers on page 79](#)

Disclaimers

This report is subject to the Terms of Use and the following disclaimers:

The information contained in this report is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied, including but not limited to the implied warranties of merchantability, non-infringement, and fitness for a particular purpose. In addition, this information is based on HCL's current product plans and strategy, which are subject to change by HCL without notice. HCL shall not be responsible for any direct, indirect, incidental, consequential, special or other damages arising out of the use of, or otherwise related to, this report or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from HCL or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of HCL software.

References in this report to HCL products, programs, or services do not imply that they will be available in all countries in which HCL operates. Product release dates and/or capabilities referenced in this presentation may change at any time at HCL's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Discrepancies found between reports and other HCL documentation sources may or may not be attributed to different publish and refresh cycles for this tool and other sources. Nothing contained in this report is intended to, nor shall have the effect of, stating or implying that any

activities undertaken by you will result in any specific sales, revenue growth, savings or other results. You assume sole responsibility for any results you obtain or decisions you make as a result of this report.

Notwithstanding the Terms of Use users of this site are permitted to copy and save the reports generated from this tool for such users own internal business purpose. No other use shall be permitted.

Hardware

You can find information about the hardware requirements for HCL DevOps Test Hub (Test Hub).

Hardware	Requirement	Notes
Disk space	256 GB	Additional resources are required by the cluster for the execution of test assets. Disk space requirements can grow in line with quantity of test assets and results stored.
Memory	32 GB	Additional resources are required by the cluster for the execution of test assets.
Processor	8 CPUs	Additional resources are required by the cluster for the execution of test assets.
Network	Gigabit (1000) Ethernet	
Other hardware		Operating system support is specified by the supported platform for Kubernetes. Refer to their associated documentation for a list of supported operating systems.

Related information

[System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#)

Operating systems and Containers

You can find details about the supported operating systems and containers for HCL DevOps Test Hub (Test Hub).

Contents

- [Bit version support on page 81](#)
- [Operating systems on page 81](#)
- [Containers on page 81](#)

Bit version support

Different parts of a product might run on the same operating system but support different application bitness.

For example, one part of the product might run only in 32-bit mode, whereas another might support 64-bit tolerate mode.

Bitness	Description
64-Tolerate	The product or part of the product runs as a 32-bit application in the 64-bit platforms listed as supported.
64-Exploit	The product or part of the product runs as a 64-bit application in the 64-bit platforms listed as supported.

Operating systems

Operating system	Hardware	Bitness	Notes
Ubuntu Server 20.04 LTS	x86-64	64-Exploit	Running K3s v1.29
Ubuntu Server 22.04 LTS	x86-64	64-Exploit	Running K3s v1.29
RHEL 8.8	x86-64	64-Exploit	Running K3s v1.29
RHEL 9.2	x86-64	64-Exploit	Running K3s v1.29

Containers

You can find details about the supported containers.

Runtime	Version
Azure Kubernetes Service	1.27

Runtime	Version
	1.28

Related information

[System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#)

Host prerequisites

You can find the prerequisites that support the operating capabilities for HCL DevOps Test Hub (Test Hub).

Contents

- [Licensing on page 82](#)

Licensing

License Server	Version
FlexNet Operations Cloud	Latest Cloud version
HCL Local License Server	3.0 also known as 2022.12

Related information

[System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#)

Supported software

You can find details about the additional software for HCL DevOps Test Hub (Test Hub) that are supported.

Contents

- [Development tools on page 83](#)
- [Integrations on page 83](#)
- [Runtime environment on page 85](#)
- [Web browsers on page 85](#)

Development tools

Supported software	Version
GitHub	2.19
	2.21
	2.22
	3.1
	3.3
	3.4
	3.5
	3.6
	HCL GitHub 3.6.10
	3.7
	3.8
HCL GitHub 3.8.12	

Integrations

Supported software	Version
Azure Plugins	Cloud version, latest
GitHub Actions	Latest
Grafana	10.x

Supported software	Version
HCL® Compass	2.2
HCL DevOps Deploy	8.0.0
HCL® Launch	7.1
	7.2.1
	7.2.3
	7.3
	7.3.2
IBM® Engineering Test Management	7.0.1
	7.0.2
	7.0.3
IBM® Rational® ClearQuest®	10.x
IBM® DevOps Deploy	8.0.0
IBM® UrbanCode™ Deploy	7.1
	7.2.1
	7.2.3
	7.3
	7.3.2

Supported software	Version
Jenkins	2.x
Jira	8.x
	9.x
Jira cloud	Cloud version, latest
Xray app for Jira	Cloud version, latest

Runtime environment

Supported software	Version	Notes
Docker	23	Optional for remote execution of test assets on Docker
	24	
Helm CLI	3.14.2	Required locally for Helm.

Web browsers

Browser	Version
Apple Safari	14 or later
Google Chrome	120 or later
Microsoft Edge	120 or later
Mozilla Firefox	120 or later
Mozilla Firefox ESR	115 or later

Related information

[System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#)

Chapter 3. Getting Started Guide

This guide provides an overview of HCL DevOps Test Hub (Test Hub). You can find the task flows to get you started with Test Hub. This guide is intended for new users.

Before you can perform the various tasks described in the *Getting Started Guide* and the other guides, you must install Test Hub. See [Installation of DevOps Test Hub on page 93](#).

Overview of DevOps Test Hub

HCL DevOps Test Hub (Test Hub) brings together test data, test environments, and test runs and reports into a single, web-based browser for testers and non-testers.

Test Hub provides the following capabilities:

Web-based continuous testing platform

Test Hub is a web-based continuous testing platform built on modern, cloud native technologies that enables test teams to run a breadth of tests that includes API, functional, and performance tests and to benefit from a holistic view of test progress. Test Hub is built on Docker for easy deployment. You can scale your testing on the cloud or remote on premise systems with native Docker, Kubernetes, and IBM® Red Hat OpenShift support.

Role-based access and security

Security is a key concern for HCL clients and therefore, Test Hub brings a comprehensive, role-based access control scheme to the server with project owners assigning key permissions (by using roles) for specific members, for example managing test data or working with secrets such as passwords.

Running of tests from the server by using Docker containers

Server-based running of tests is the starting point for Test Hub. For members of a project with the appropriate role, Test Hub enables direct running of tests from the browser by using transient Docker containers.

Connected agents for existing performance agents

Agent owners can connect existing performance agents to the server and add them to a project for running schedules and Accelerated Functional Testing (AFT) Suites on the current infrastructure.

Project overview statistics

The **Overview** page for Test Hub offers you a quick, simple view on the state of testing for your projects.

Project home page

The home page lists your projects and other projects, which makes it easy to manage different projects within an organization.

Reporting and the Resource Monitoring Service

Test Hub provides the home for capabilities that previously were hosted on HCL DevOps Test Virtualization Control Panel (Test Virtualization Control Panel). Reporting and the Resource Monitoring

Service are in Test Hub and provide a more direct relationship with their related projects. These capabilities also benefit from the project level, role-based access controls. You can view unified test results to help you make informed business decisions.

Test data authoring

Beyond the concept of a project held in a Git repository for a simple location of tests and related assets, you can do full concurrent editing of test data sets directly from Test Hub. This true multiuser capability enables team members to collaborate more easily as well as try out data changes without impacting the rest of the team. When satisfied with the results, team members can push their changes. You can also fabricate sample data to perform tests by using the data generator supported in Test Hub.

Integration with DevOps tools

You can integrate Test Hub with various popular DevOps tools like Jenkins, DevOps Deploy, or Microsoft Azure DevOps to get more value from your DevOps pipelines. You can use Deploy to define a deployment process that automatically triggers test cases and have those test insights available directly within HCL DevOps Velocity (Velocity). With the Microsoft Azure DevOps integration, you can also define a Azure DevOps pipeline that includes direct execution of tests by using the Test Hub scalable infrastructure. Test Hub also integrates with Azure DevOps, Jira, Rational® ClearQuest® or HCL® Compass for defect tracking and GitHub for software source management and version control.

Supported versions of assets and resources

You can find information about the versions of assets, resources, agents and Dockers that are supported in HCL DevOps Test Hub (Test Hub). You can also find information about the versions of the desktop clients that are supported.

The following table lists the versions of assets or resources created in the desktop clients along with the agents, Dockers, or proxies that are supported in Test Hub:

Resource	Supported	Not supported
Assets or resources created in HCL DevOps Test UI (Test UI)	Assets or resources created in Test UI that is of a version earlier than or the same as Test Hub.	Assets or resources created in any later version of Test UI that is later than the version of Test Hub.
Assets or resources created in HCL DevOps Test Integrations and APIs (Test Integrations and APIs)	Assets or resources created in Test Integrations and APIs that is of a version earlier than or the same as Test Hub.	Assets or resources created in any later version of Test Integrations and APIs that is later than the version of Test Hub.
Assets or resources created in HCL DevOps Test Performance (Test Performance)	Assets or resources created in Test Performance that is of a version earlier than or the same as Test Hub.	Assets or resources created in any later version of Test Performance that is later than the version of Test Hub.

Resource	Supported	Not supported
HCL DevOps Test Performance Agent (Test Performance Agent)	Test Performance Agent, which is the same version as Test Hub.	Before you install the agents that are of the current version, see Management of performance agents on page 386 .
HCL DevOps Test Integrations and APIs Agent (Test Integrations and APIs Agent)	Test Integrations and APIs Agent, which is the same version as Test Hub.	Before you install the agents that are of the current version, see Management of API agents on page 394 .
Dockers on a remote host computer	Image of Test Hub, which is the same version as Test Hub that you want to use to run tests, must be installed on the Docker on a remote host computer.	Before you set up Dockers remotely, see Management of Docker hosts on page 401 .
HTTP proxies	Installation of the proxy component from Test Virtualization Control Panel, which is the same version as Test Hub.	Before you use HTTP proxies, see Setting up intercepts on page 744 .

If there is a mismatch in the versions of the assets, resources, agents, or images used in Dockers with the version of Test Hub that you are using, any of the following events can occur:

- Warnings or Errors are displayed when you add a repository to a project in your team space.
- Warnings or Errors are displayed when you open a project that you migrated from a previous version of Test Hub.
- Warnings or Errors are displayed when you add or manually refresh a repository to which you added assets or resources that were created on a desktop client with a version previous to or later than the version of Test Hub that you want to use.
- Test runs that you start or schedule on an agent or Docker fail to run.
- You cannot view the proxies that are registered with Test Hub on the **Agents and Intercepts** page.

Suites and tests that are supported on DevOps Test Hub

You can find information about the types of Test Suites and tests that you can configure for a run in a project on HCL DevOps Test Hub (Test Hub).

The following table lists the Test Suites and test types that are supported and those tests that are not supported to run on Test Hub:

Supported Suites	Supported tests	Unsupported tests
<ul style="list-style-type: none"> • Accelerated Functional Testing (AFT) Suites • Compound Tests that contain any of the following types of tests: <ul style="list-style-type: none"> ◦ Web UI tests ◦ HTML tests ◦ Mobile tests 	<ul style="list-style-type: none"> • Individual Web UI test • Individual Windows test • Individual Mobile test 	<ul style="list-style-type: none"> • Traditional HTML tests • Traditional non-HTML tests
<ul style="list-style-type: none"> • API Suites that include the following Test Suites: <ul style="list-style-type: none"> ◦ Test Suites that contain scenarios with references to local stubs. ◦ Test Suites that contain stubs that virtualize services run in any Kubernetes cluster or in the Kubernetes cluster that hosts Test Hub. • Virtual services or Stubs 	API tests	Suites or tests with subscribe actions that operate in the watch mode.
<ul style="list-style-type: none"> • Compound Tests that contain performance tests • Rate Schedules • VU Schedules 	Individual Performance test	<ul style="list-style-type: none"> • A single test script of any test extension. • Tests that belong to 32-bit test extensions and SOA Quality included in VU Schedule, Rate Schedule, and Compound Test. • Citrix, SAP, or Siebel as an individual test or as part of a Compound Test.
Postman resources		
JMeter tests		
JUnit tests		

Supported Suites	Supported tests	Unsupported tests
Selenium tests		



Note: You must create Compound Tests for the same type of tests. For example, you can create a Compound Test that has different mobile tests that use different mobile testing platforms.



Restriction: You cannot merge folders that contain different test projects in the same project directory. For example, you cannot create a project directory that contains a Performance test project and an API Suite project. You must create separate project directories for each test type. For example, you must create the Performance test in the project directory as `/MyPerfProject/` and the API Suite in the project directory as `/MyAPIProject`, before you commit the projects to the remote repository.

Related information

[Test run configurations on page 446](#)

Virtual services that are supported on DevOps Test Hub

You can find information about the virtual services that are supported on HCL DevOps Test Hub (Test Hub).

You can start a running instance of the virtual services from Test Hub to run on the following locations:

- A registered API agent.
- The default Kubernetes cluster of Test Hub.
- A remote Kubernetes cluster.

You can run all virtual services on registered API agents except the virtual services that use CICS transport, DB2/Z transport, or Istio.

The default Kubernetes cluster or a remote Kubernetes cluster supports running of the following types of virtual services:

- Virtual services that utilize the WebSphere® MQ transport.
- Virtual services that utilize the HTTP transport.



Restriction: You cannot run virtual services on a remote Docker host.

Related information

[Management of virtualized services on page 741](#)

Accessibility features of DevOps Test Hub

Accessibility features help users who have physical disabilities, such as visual and, hearing impairment, or limited mobility, to use the software products successfully.

Accessibility compliance

The product documentation is published by using Oxygen XML WebHelp Responsive. To understand the accessibility compliance status for Oxygen XML WebHelp Responsive, refer to [WebHelp Responsive VPAT Accessibility Conformance Report](#).

Accessing UI elements

Test Hub supports navigation in the UI by using different methods such as a mouse, keyboard, or touchpad.

You can use the keyboard keys such as **Tab**, arrow keys such as **UP**, **DOWN**, **LEFT**, and **RIGHT** to navigate to the different pages in the **Navigation** pane or to the different action labels in the right pane on the UI.

Chapter 4. Administrator Guide

This guide describes how to install the HCL DevOps Test Hub (Test Hub) software.

After you install the software, you can perform administration tasks such as license configuration, user management, security, memory and disk usage management, back up and restore user data, and other tasks that a server administrator can perform. This guide is intended for administrators.

As a *Server Administrator*, you can assign the role of a *Server Administrator* to any user. For information about how to change the role, see [Default user administration on page 139](#).



Remember: When your role is changed from a user to a *Server Administrator*, you must log out and log in again so that Test Hub can apply the change in your role.

Installation of DevOps Test Hub

To get started working with HCL DevOps Test Hub (Test Hub), you must first install Test Hub.

You can install Test Hub on the following platforms:

- Azure Kubernetes Service (AKS)
- RHEL Server (by using k3s)
- Ubuntu Server (by using k3s)

Installation task flows

Azure Kubernetes Service

You must perform the following tasks to install Test Hub:

1. Install Azure command-line interface (Azure CLI) software and connect to your Azure subscription.
2. Install kubectl.
3. Install the Helm software.
4. Install Emissary-ingress.
5. Install Test Hub.

RHEL Server

You must perform the following tasks to install Test Hub:

1. Set up RHEL Server.
2. Install the OpenSSH server and the Helm software.
3. Install Test Hub.

Ubuntu Server

You must perform the following tasks to install Test Hub:

1. Set up Ubuntu Server.
2. Install the OpenSSH server and the Helm software.
3. Install Test Hub.

To learn more about the installation of Test Hub on platforms such as AKS, RHEL, or Ubuntu refer to the following topics:

Installation of DevOps Test Hub on Azure Kubernetes Service

You can find information about the tasks that you can perform to install HCL DevOps Test Hub (Test Hub) on the Azure Kubernetes Service (AKS) platform.

Prerequisites for installing DevOps Test Hub on Azure Kubernetes Service

You must complete certain tasks before you install HCL DevOps Test Hub (Test Hub) on the Azure Kubernetes Service (AKS) platform.

The following sections describe each prerequisite in detail:

- [Internet access on page 94](#)
- [Azure subscription on page 94](#)
- [Azure Kubernetes Service on page 95](#)
- [Mandatory software on page 95](#)
- [Azure resource provider on page 96](#)
- [VPN Certificate on page 96](#)

Internet access

You must have access to the internet to install Test Hub.

Azure subscription

You must have an active Azure subscription along with the following Azure built-in roles:

Categories	Roles	Go to...
Azure built-in	Owner	Azure built-in roles in Azure Kubernetes Service documentation.

Categories	Roles	Go to...
	Or Contributor and User Access Administrator	



Note: You can create a subscription from [Microsoft Azure](#) if you do not have a valid subscription.

Azure Kubernetes Service

You must ensure that you select the size of the virtual machine as *Standard_D8ds_v4* while setting up the AKS cluster.

For more information about specific versions of software requirements, see [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).

For more information about Azure Kubernetes Service and deployment options, refer to [Azure Kubernetes Service](#) documentation.

Mandatory software

You must install the following mandatory software:

- Git Bash. For more information, refer to the [Git](#) documentation.
- Helm. For more information, refer to the [Helm](#) documentation. For more information about specific versions of software requirements, see [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).



Note: The Helm command must be in one of the directories in your **PATH** environment variable.

- Azure command-line interface (Azure CLI) 2.56.0. You can upgrade using the following command:

```
az upgrade --yes
```

For more information, refer to the [Azure CLI](#) documentation.



Note: You can run the `az version` command to find the version and dependent libraries that are installed.

- Azure Bicep CLI version 0.24.24. You can upgrade using the following command:

```
az bicep upgrade
```

- kubectl tool. For more information, refer to the [Kubernetes](#) documentation.
- Emissary-ingress. See [Installing Emissary-ingress on page 97](#).

Azure resource provider

You must register an Azure resource provider so that the AKS cluster can provision to use Persistent Volumes with Azure Files for the ReadWriteOnceMany (RWX) storage class. You can run the following command to enable the storage for the RWX storage class:

```
az provider register -n Microsoft.Storage
```

For more information about Azure resource providers and types, refer to [Azure Kubernetes Service](#) documentation.

VPN Certificate

Create a root and client certificate and name it as `P2SRootCert.cer`. Refer to the [documentation](#)

Export the root certificate and save the certificate in the same folder as `main.bicep`.

Related information

[Emissary-ingress](#)

Cluster autoscaler on Azure Kubernetes Service

You can enable the cluster autoscaler feature on Azure Kubernetes Service (AKS) to manage the cost of the cluster effectively.

You can use the cluster autoscaler with multiple node pools that are enabled. Depending on your AKS setup, when the nodes are not in use, the nodes of the AKS cluster are automatically shut down to minimize the runtime costs of the AKS cluster without risking the performance of HCL DevOps Test Hub (Test Hub).

When you create an AKS cluster, that contains a single node pool as the default node pool. You can use the `az aks node pool` command to add a node pool to your existing AKS cluster. You can then use the `az aks update` command to enable and configure the cluster autoscaler on the node pool for the existing AKS cluster.

You can use the following commands to add a node pool and enable the cluster autoscaler on the node pool for the existing AKS cluster:

```
az aks nodepool add -g <resource_group> -n wrk0 \
  --cluster-name <aks_cluster_name> \
  --kubernetes-version <k8s_version> \
  --node-vm-size Standard_D2ds_v4 \
  --enable-cluster-autoscaler \
  --node-count 1 --min-count 1 --max-count 2 \
  --labels execution=allow \
  --node-taints reserved=reserved:NoSchedule

az aks update \
  -g <resource_group> -n <aks_cluster_name> \
  --cluster-autoscaler-profile scale-down-unnneeded-time=1m scale-down-delay-after-add=1m
```



Note: You must replace `<resource_group>`, `<k8s_version>`, and `<aks_cluster_name>` with the name of the resource group, the version number of Kubernetes, and the name of the AKS cluster in the command.

After you run the commands, you have completed the following actions:

- Added a user node pool called `wrk0`
- Enabled the cluster autoscaler feature
- Updated the default values of **scale-down-unnneeded-time** and **scale-down-delay-after-add** in the cluster-wide autoscaler profile to `1m`

After you install Test Hub by enabling the cluster autoscaler feature on your AKS cluster, a new container is used only to run test assets. You can run the `kubectl get pods` command to view the status of the pod, which is displayed as `Pending`.

You can describe the pod by running the `kubectl describe pod <pod_name> -n <namespace>` command to view a scale event when you run test assets from Test Hub.

When the test asset run is complete, the pod shuts down automatically. Thus, you can reduce the cost of the cluster when the node is not in use. You can then run the `kubectl get nodes` command to verify the number of nodes that are currently in the `Ready` state.

Related information

[Automatically scale a cluster to meet application demands on Azure Kubernetes Service \(AKS\)](#)

[Create and manage multiple node pools for a cluster in Azure Kubernetes Service \(AKS\)](#)

Installing Emissary-ingress

You must install Emissary-ingress to terminate Transport Layer Security (TLS) and route traffic to HCL DevOps Test Hub (Test Hub). You can install Emissary-ingress by pulling the images from Azure Container Registry (ACR) when you do not want to pull the images from the public registries.

Before you begin

You must have completed the following tasks:

- Set up the Azure Kubernetes Service (AKS) cluster with Kubernetes. For more information, refer to [Azure Kubernetes Service](#) documentation.
- Installed the following software:

- Git Bash. For more information, refer to the [Git](#) documentation.
- Helm. For more information, refer to the [Helm](#) documentation. For more information about specific versions of software requirements, see [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).



Note: The Helm command must be in one of the directories in your **PATH** environment variable.

- kubectl tool. For more information, refer to the [Kubernetes](#) documentation.
- Azure command-line interface (Azure CLI) 2.56.0. You can upgrade using the following command:

```
az upgrade --yes
```

For more information, refer to the [Azure CLI](#) documentation.



Note: You can run the az version command to find the version and dependent libraries that are installed.

- Azure Bicep CLI version 0.24.24. You can upgrade using the following command:

```
az bicep upgrade
```

- Logged in to the AKS cluster.



Note: You can run the az login command to log in to AKS interactively.

- Subscribed to active Azure subscription.



Note: You can run the az account set --subscription {subscription_name} command to set your active subscription.

- Configured the location of your AKS cluster.



Note: You can run the az configure --defaults location={location_name} command to set the location of your AKS cluster.




Tip: You can run the following command to get the list of all the available locations:

```
az account list-locations --query "[].{DisplayName:displayName, Name:name}" -o table
```

About this task

The following table lists the variables that you must replace with the actual value in the command.

Variables	Description
{resource_group}	The name of the resource group that you created during the creation of the AKS cluster.
{azure_cluster}	The name of the Azure cluster that you created during the creation of the AKS cluster.
{azure_container_registry}	The name of the ACR that you created during the AKS setup.
{ip_address}	The internal IP address to use with the Emissary-ingress.  Note: You must ensure that the IP address is not in use within your virtual network.

1. Run the following command to configure kubectl to connect to the AKS cluster:

```
az aks get-credentials -g {resource_group} -n {azure_cluster} --public-fqdn --overwrite-existing
kubectl get nodes
```



Note: You can verify the connection to the AKS cluster by running the `kubectl get nodes` command that displays a list of the cluster nodes.

2. Run the following command to pull images into ACR:

```
az acr import -n {azure_container_registry} -g {resource_group} \
  --source docker.io/emissaryingress/emissary:3.5.1 \
  --image emissaryingress/emissary:3.5.1
```



Remember: `emissary-ingress` is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of the `emissary-ingress` in all the instances in this procedure.

3. Run the following commands to install the Emissary-ingress:

```
kubectl create namespace emissary 2>/dev/null || true
sed -e "s#docker.io/#{azure_container_registry}.azurecr.io/#"
  ibm-rtas-prod/files/azure/manifests/emissary-crds.yaml \
  | kubectl apply -f -
kubectl wait deployment emissary-apiext --for condition=available -n emissary-system
--timeout=90s
sed -e "s#docker.io/#{azure_container_registry}.azurecr.io/#; s/\$INGRESS_IP/\$INGRESS_IP/"
  ibm-rtas-prod/files/azure/manifests/emissary-emissaryns-azure.yaml \
  | kubectl apply -f -
```

```
kubectl rollout status -n emissary deployment/emissary-ingress -w
```

4. Run the following command to verify that external-IP is assigned to Emissary-ingress:

```
kubectl get svc -n emissary -w emissary-ingress
```



Note: The status of EXTERNAL-IP is displayed as `Pending` for a few minutes. If the status does not change to `{ip_address}`, then that indicates you have a permission issue.

You can run the following command to investigate the issue:

```
kubectl describe svc -n emissary emissary-ingres
```

Results

You have installed Emissary-ingress.

What to do next

You can install Test Hub. See [Installing DevOps Test Hub on Azure Kubernetes Service on page 100](#).

Related information

[Create an ingress controller to an internal virtual network in Azure Kubernetes Service \(AKS\)](#)

[Create an ingress controller to an internal virtual network in Azure Kubernetes Service \(AKS\)](#)

[Emissary-ingress documentation](#)

Installing DevOps Test Hub on Azure Kubernetes Service

You can install HCL DevOps Test Hub (Test Hub) on Azure Kubernetes Service (AKS) that has a Kubernetes environment to run functional, integration, and performance tests. Test Hub combines all the capabilities into a single web-based browser for testers and non-testers. The capabilities include test runs, test data, test environment, or test reports.

Before you begin

- You must have performed the following tasks:
 - Completed the tasks provided in the Prerequisites section. See [Prerequisites for installing DevOps Test Hub on Azure Kubernetes Service on page 94](#).
 - **Optional:** Enabled the autoscaler feature if you want a new container to be used only to run test assets. See [Cluster autoscaler on Azure Kubernetes Service on page 96](#).
 - Logged in to the AKS cluster.



Note: You can run the `az login` command to log in to AKS interactively.

- Subscribed to active Azure subscription.



Note: You can run the `az account set --subscription {subscription_name}` command to set your active subscription.

- Configured the location of your AKS cluster.



Note: You can run the `az configure --defaults location={location_name}` command to set the location of your AKS cluster.






Tip: You can run the following command to get the list of all the available locations:

```
az account list-locations --query "[].{DisplayName:displayName, Name:name}" -o table
```

About this task

The following table lists the variables that you must replace with the actual value in the command.

Variables	Description
{RESOURCE_GROUP}	The name of the resource group that you created during the creation of the AKS cluster.
{azure_cluster}	The name of the Azure cluster that you created during the creation of the AKS cluster.
{azure_container_registry}	The name of the ACR that you created during the AKS setup.
{my-ingress-dns-name}	The Ingress name and DNS zone value that you provided during the setting up of the AKS cluster. For example, if the Ingress name and DNS zone value that you provided are <code>devops, my-org.nonprod</code> , then <code>{my-ingress-dns-name}</code> value must be <code>devops.myorg.nonprod</code> .
<code>\$HELM_NAME</code>	The release name of your choice.

Variables	Description
	 Note: The release name must consist of alphanumeric characters that are in lowercase or "-" (hyphen). The release name must also start with an alphabetic character and end with an alphanumeric character. For example, <i>my-org</i> or <i>abc-123</i> .
{HCL_LICENSING_ID}	<p>The ID of the License Server for the initial team space, if you want to set the license for the first time.</p>  Important: When you want to upgrade the product from the previous version, you must configure the value of License Server ID from the Team Space License Configuration page when the installation of the server is complete.
{PASSWORD_SEED}	<p>A value of your choice for the password.</p>  Important: This password seed is used to create several default passwords for the server. You must store the password seed securely. When you install Test Hub by using the backup of the user data, you can reuse the password seed. You can use this seed to restore the backed-up files either on the current or later versions of Test Hub.
{ip_address}.nip.io	<p>The IP address that is assigned to Emissary-ingress.</p> <p>You can run the following command to obtain the IP address of Emissary-ingress:</p> <pre>kubectl get ingress -A</pre>



Remember: The `devops-system` is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of `devops-system` in all the instances in this procedure.

1. Run the following command:

```
helm pull --untar oci://hclcr.io/ot/hcl-devops --version 11.0.1
cd hcl-devops
```

2. Run the following command to configure kubectl to connect to the AKS cluster:

```
az aks get-credentials -g {RESOURCE_GROUP} -n {INSTANCE} --public-fqdn --overwrite-existing
kubectl get nodes
```



Note: You can verify the connection to the AKS cluster by running the `kubectl get nodes` command that displays a list of the cluster nodes.

3. Run the following command to install Test Hub:



Notes:

- You must provide the namespace as the value for `$NAMESPACE` in the command. You can use `devops-system` as the value for `$NAMESPACE`.

You must provide the value for `$HELM_NAME` in the command. You can use `main` as the value for `$HELM_NAME`.

- If you enabled the autoscaler feature to use a new container only to run test assets, then you must add `-f hcl-devops/values-dedicated-nodes.yaml` in the following helm install command.
- When the ingress domain is accessible to untrusted parties, then you must set the `signup` parameter to `false` in the following command.

```
helm update --install $HELM_NAME . -n $NAMESPACE \
  --create-namespace \
  --set global.domain=$INGRESS_DOMAIN \
  -f values-k8s.yaml \
  -f hcl-devops/values-dedicated-nodes.yaml \
  --set global.persistence.rwxStorageClass=azurefile \
  --set imageRegistry=$IMAGE_REGISTRY \
  --set-literal passwordSeed=$PASSWORD_SEED \
  --set signup=true \
  --set hclFlexnetURL=https://hclsoftware.compliance.flexnetoperations.com \
  --set hclFlexnetID=$HCL_LICENSING_ID
```

4. Run the following command from the `hcl-devops/files` directory to verify and test the installed server software:

```
bash hcl-devops/files/helm-test-diag.sh {main} -n devops-system
```

Results

You have installed Test Hub. The terminal displays the following information:

- The URL to access Keycloak to manage and authenticate users.

The username can be `keycloak` and the password can be retrieved by running the following command:

```
kubectl get secret -n devops-system {main} -keycloak-postgresql -o jsonpath="{.data.password}" |
base64 --decode; echo
```

Where:

- {main} is the name of the release that was provided during the installation of Test Hub.
- devops-system is the name of the namespace that was created during the installation of Test Hub.
- The URL to access the Test Hub UI.

What to do next

You can perform the following tasks:

- Configure licenses for team spaces. See [Configuring licenses for team spaces on page 181](#).
- Configure Test Hub. See [Configuration of DevOps Test Hub on page 137](#).

Related information

[Helm documentation](#)

[Azure Kubernetes Service documentation](#)

Installation of DevOps Test Hub on RHEL

You can find information about the tasks that you can perform to install HCL DevOps Test Hub (Test Hub) software on the RHEL platform.

Prerequisites to install DevOps Test Hub on RHEL

You must complete certain tasks before you install HCL DevOps Test Hub (Test Hub) on the RHEL platform.

The following sections describe the prerequisites in detail:

- [Internet access on page 104](#)
- [RHEL Server on page 104](#)
- [Mandatory software on page 105](#)

Internet access

You must have access to the internet to install Test Hub.

RHEL Server

You must install the supported version of RHEL Server and have a working Domain Name Server (DNS) that resolves the hostname into a machine-readable IP address.

For more information about specific versions of software requirements, see [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).



Note: Depending on your testing workload, Test Hub might require more resources. You must use the entire disk space and set up Logical Volume Manager (LVM) by using the `ext4` file system. If your organization requires user data to be stored in a separate partition, then you can create a mount point at `/var/lib/rancher/k3s/storage/` with at least 128 GB capacity.

Mandatory software

You must install the following mandatory software:

- Helm. For more information, refer to the [Helm](#) documentation. For more information about specific versions of software requirements, see [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).



Note: The Helm command must be in one of the directories in your **PATH** environment variable.

- OpenSSH Server. For more information, refer to the Installation section in the [OpenSSH Server](#) documentation.



Note: If you use ssh to log in to the computer, then you can skip this step because OpenSSH is already installed on your computer.

Installing DevOps Test Hub on RHEL


You can install HCL DevOps Test Hub (Test Hub) on the RHEL server. Test Hub combines test data, test environments, test runs and reports into a single, web-based report for testers and non-testers.



Before you begin

- You must have read and been familiar with additional Helm parameters. See [Additional Helm parameters on page 111](#).
- You must have performed the following tasks:
 - Completed the tasks provided in the Prerequisites section. See [Prerequisites to install DevOps Test Hub on RHEL on page 104](#).

About this task

The following table lists the variables that you must replace with the actual value in the command.

Variables	Description
{PASSWORD_SEED}	<p>A value of your choice for the password.</p> <p> Important: This password seed is used to create several default passwords for the server. You must store the password seed securely. When you install Test Hub by</p>

Variables	Description
	 using the backup of the user data, you can reuse the password seed. You can use this seed to restore the backed-up files either on the current or later versions of Test Hub.
{HCL_LICENSING_ID}	<p>The ID of the License Server for the initial team space, if you want to set the license for the first time.</p>  Important: When you want to upgrade the product from the previous version, you must configure the value of License Server ID from the Team Space License Configuration page when the installation of the server is complete.

1. Log in to the Ubuntu server using an SSH session.
2. Run the following command:

```
helm pull --untar oci://hclcr.io/ot/hcl-devops --version 11.0.1
cd hcl-devops
```

3. Run the following command to install Test Hub:



Notes:

- You must provide the namespace as the value for `$NAMESPACE` in the command. You can use `devops-system` as the value for `$NAMESPACE`.

You must provide the value for `$HELM_NAME` in the command. You can use `main` as the value for `$HELM_NAME`.

- When the ingress domain is accessible to untrusted parties, then you must set the `signup` parameter to `false` in the following command.

```
chmod +x k3s/*.sh

sudo k3s/init.sh -n $NAMESPACE \
  --set-literal passwordSeed=$PASSWORD_SEED \
  --set signup=true \
  --set hclFlexnetURL=https://hclsoftware.compliance.flexnetoperations.com \
  --set hclFlexnetID=$HCL_LICENSING_ID
```

4. **Optional:** Run the following command to remove a job that is used to initialize the PostgreSQL database during the installation of Test Hub:

```
kubectl delete job {{main}} -postgresql-init -n devops-system
```

5. Run the following script from the `hcl-devops/files` directory to verify and test the installed instance:

```
hcl-devops/files/helm-test-diag.sh {main} -n devops-system
```

Results

On completion of the installation of server software, the output displays the following information on the command-line interface:

- Instructions to access Keycloak to manage and authenticate users.

The username can be `keycloak` and the password can be retrieved by running the following command:

```
kubectl get secret -n devops-system -keycloak-postgresql -o jsonpath="{.data.password}" | base64 --decode; echo
```

- The URL to access the Test Hub UI.

What to do next

You can perform the following tasks:

- Configure licenses for team spaces. See [Configuring licenses for team spaces on page 181](#).
- Configure Test Hub. See [Configuration of DevOps Test Hub on page 137](#).

Related information

[Helm documentation](#)

[Troubleshooting on page 1077](#)

Installation of DevOps Test Hub on Ubuntu

You can find information about the tasks that you can perform to install HCL DevOps Test Hub (Test Hub) software on the Ubuntu platform.

Prerequisites to install DevOps Test Hub on Ubuntu

You must complete certain tasks before you install HCL DevOps Test Hub (Test Hub) on the Ubuntu platform.

The following sections describe prerequisites in detail:

- [Internet access on page 108](#)
- [Ubuntu Server on page 108](#)
- [Backup of user data on page 108](#)
- [Mandatory software on page 108](#)

Internet access

You must have access to the internet to install Test Hub.

Ubuntu Server

You must install the supported version of Ubuntu Server and have a working Domain Name Server (DNS) that resolves the hostname into a machine-readable IP address.

For more information about specific versions of software requirements, see [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).



Note: Depending on your testing workload, Test Hub might require more resources. You must use the entire disk space and set up Logical Volume Manager (LVM) by using the `ext4` file system. If your organization requires user data to be stored in a separate partition, then you can create a mount point at `/var/lib/rancher/k3s/storage/` with at least 128 GB capacity.

Backup of user data

If you want to upgrade Test Hub from a previous version, then you must perform one of the following tasks based on the existing version of Test Hub:

- Back up the data from V10.0.2, Fix Pack 1 or earlier. See [Backing up the user data from a previous release](#).
- Back up the data from V10.1.0 or later. See [Backing up DevOps Test Hub data on Ubuntu on page 126](#).



Note: You must select the existing version of Test Hub from the drop-down list to view the procedure to back up and restore the user data for your version because the instructions differ for different versions.

Mandatory software

You must install the following mandatory software:

- Helm. For more information, refer to the [Helm](#) documentation. For more information about specific versions of software requirements, see [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).



Note: The Helm command must be in one of the directories in your **PATH** environment variable.

- OpenSSH Server. For more information, refer to the Installation section in the [OpenSSH Server](#) documentation.



Note: If you use ssh to log in to the computer, then you can skip this step because OpenSSH is already installed on your computer.

Installing DevOps Test Hub on Ubuntu


You can install HCL DevOps Test Hub (Test Hub) on the Ubuntu server. Test Hub combines test data, test environments, and test runs and reports into a single, web-based browser for testers and non-testers.


Before you begin

- You must have read and been familiar with additional Helm parameters. See [Additional Helm parameters on page 111](#).
- Completed the tasks provided in the Prerequisites section. See [Prerequisites to install DevOps Test Hub on Ubuntu on page 107](#).

About this task

The following table lists the variables that you must replace with the actual value in the command.

Variables	Description
{PASSWORD_SEED}	<p>A value of your choice for the password.</p> <p> Important: This password seed is used to create several default passwords for the server. You must store the password seed securely. When you install Test Hub by using the backup of the user data, you can reuse the password seed. You can use this seed to restore the backed-up files either on the current or later versions of Test Hub.</p>
{HCL_LICENSING_ID}	The ID of the License Server for the initial team space, if you want to set the license for the first time.

Variables	Description
	 Important: When you want to upgrade the product from the previous version, you must configure the value of License Server ID from the Team Space License Configuration page when the installation of the server is complete.

1. Log in to the Ubuntu server using an SSH session.
2. Run the following command:

```
helm pull --untar oci://hclcr.io/ot/hcl-devops --version 11.0.1
cd hcl-devops
```

3. Run the following command to install Test Hub:



Notes:

- You must provide the namespace as the value for `$NAMESPACE` in the command. You can use `devops-system` as the value for `$NAMESPACE`.

You must provide the value for `$HELM_NAME` in the command. You can use `main` as the value for `$HELM_NAME`.

- When the ingress domain is accessible to untrusted parties, then you must set the `signup` parameter to `false` in the following command.

```
chmod +x k3s/*.sh

sudo k3s/init.sh -n $NAMESPACE \
  --set-literal passwordSeed=$PASSWORD_SEED \
  --set signup=true \
  --set hclFlexnetURL=https://hclsoftware.compliance.flexnetoperations.com \
  --set hclFlexnetID=$HCL_LICENSING_ID
```

4. **Optional:** Run the following command to remove a job that is used to initialize the PostgreSQL database during the installation of Test Hub:

```
kubectl delete job {{main}} -postgresql-init -n devops-system
```

5. Run the following script from the `hcl-devops/files` directory to verify and test the installed instance:

```
hcl-devops/files/helm-test-diag.sh {main} -n devops-system
```

Results

On completion of the installation of server software, the output displays the following information on the command-line interface:

- Instructions to access Keycloak to manage and authenticate users.

The username can be `keycloak` and the password can be retrieved by running the following command:

```
kubectl get secret -n devops-system -keycloak-postgresql -o jsonpath="{.data.password}" | base64
--decode; echo
```

- The URL to access the Test Hub UI.

What to do next

You can perform the following tasks:

- Configure licenses for team spaces. See [Configuring licenses for team spaces on page 181](#).
- Configure Test Hub. See [Configuration of DevOps Test Hub on page 137](#).

You can migrate the user data if you upgraded Test Hub from the earlier version to a new version.

The migration procedure is different based on the existing version of Test Hub. For more information about migration process, go to the following topics:

- [Migrating user data on Ubuntu from 10.1.2 or earlier on page 122](#).
- [Migrating user data on Ubuntu from 10.1.3 or later on page 123](#).

Related information

[Helm documentation](#)

[Troubleshooting on page 1077](#)

Management of DevOps Test Hub features

After you install HCL DevOps Test Hub (Test Hub), you can manage certain features based on your requirements.

During the installation of Test Hub, if you did not use any Helm parameters, you can still use those parameters even after installation is complete.


You can use the helm upgrade command to enable or disable server features after the installation of Test Hub is complete.



Similarly, you can use the other additional Helm parameters to enable or disable features based on your requirements. See [Additional Helm parameters on page 111](#).

Additional Helm parameters

You can find the information about Helm parameters that you can use during the installation of HCL DevOps Test Hub (Test Hub).

As a system administrator when you install or upgrade Test Hub, you can pass values for some of the individual parameters by using `--set`. The following table lists such parameters that you can use while running the helm install or helm upgrade command to enable or disable certain features in Test Hub.

Parameters	Values	Description
keycloak.signup-enabled	true false	<p>Use this parameter to show or hide the Sign-Up link on the Test Hub Login page.</p> <p>The default value of the keycloak.signup.enabled parameter is set to <code>false</code> to enhance the security of Test Hub.</p> <p> Note: You can use this parameter only during the installation of Test Hub.</p> <p>If you have administrator access to Keycloak, then you can also enable the self-registration by using the Keycloak UI after the installation of Test Hub is complete. For more information, refer to the Keycloak documentation.</p>
router.allowedOrigin	IP address or domain name of other servers.	<p>Use this parameter to integrate call requests from other web servers to Test Hub.</p> <p>The default configuration of Test Hub rejects all Cross-origin resource sharing (CORS). Therefore, you must use the router.allowedOrigin parameter to restrict CORS to a specific whitelist.</p> <p>For example,</p> <pre>--set router.allowedOrigin=http://http.example.com:*,https:// https.mydomain.net:*</pre> <p>Where, <code>http.example.com</code> and <code>https.mydomain.net</code> are the domains that you want to provide access to Test Hub.</p> <p>You can also provide the value of router.allowedOrigin as the IP address when you cannot resolve a specific Domain Name Server (DNS). If your domain can be accessible through IP address on a specific port, then you can provide the value of router.allowedOrigin as follows:</p> <pre>--set router.allowedOrigin=*://ip-address:*,*://ip-address:port_number</pre> <p>For example,</p> <pre>--set router.allowedOrigin="*://192.0.2.0:*" --set router.allowedOrigin="*://192.0.2.1:3000"</pre>

Parameters	Values	Description
		 Note: You can provide multiple domain names or IP addresses separated by a comma (,).
networkPolicy.egress.enabled	true false	<p>Set this parameter value to <code>true</code> to restrict traffic from Test Hub to private IP addresses.</p> <p>The default configuration of Test Hub does not create any Egress policy to restrict the other endpoints that can connect with Test Hub. Therefore, you must use the <code>networkPolicy.egress.enabled</code> parameter to enable the Egress policy that can restrict traffic to private IP addresses.</p> <p>For example,</p> <pre>--set networkPolicy.egress.enabled=true</pre> <p>You can inspect the deployed policy by running the following command:</p> <pre>kubectl get networkpolicy -n {namespace} {{main}}-allow-egress -oyaml</pre>  Note: You must substitute <code>{namespace}</code> with the name of the namespace and <code>{{main}}</code> with the release name that you used during the installation of Test Hub.
execution.template.resources.limits.memory	<size_of_memory>Gi	<p>Use this parameter to run VU schedules that contain multiple Web UI tests or a large number of virtual users or a combination of Web UI, performance, and API tests.</p> <p>The default value of the <code>execution.template.resources.limits.memory</code> parameter is set to <code>3Gi</code>.</p> <p>For example,</p> <pre>--set execution.template.resources.limits.memory=5Gi</pre>

Related information

[Installing DevOps Test Hub on Ubuntu on page 109](#)

[Installing DevOps Test Hub on Azure Kubernetes Service on page 100](#)

[Management of DevOps Test Hub features on page 111](#)

DevOps Test Hub upgrade methods

When you want to use the enhanced functionalities of HCL DevOps Test Hub (Test Hub), you must upgrade Test Hub to the latest version.

You can upgrade to Test Hub from HCL OneTest™ Server 10.5.4 or 10.5.3 by using the helm update command. If you want to upgrade from versions previous to 10.5.3, you must first upgrade to 10.5.3 or 10.5.4.

You can upgrade Test Hub by using following methods:

- **In-place upgrade:** In this method, you can upgrade Test Hub in the same location that contained 10.1.3 or later version of Test Hub by using the helm upgrade command.
- **Upgrade (by new install):** In this method, you can upgrade Test Hub by uninstalling the existing version of Test Hub, and then by installing the latest version of Test Hub.

 **Important:** On the Ubuntu platform, the **In-place upgrade** method is not available. You must use the **Upgrade (by new install)** method to upgrade Test Hub.

The upgrade procedures are different based on the existing version of Test Hub. To learn more about Test Hub upgrade methods on platforms such as Ubuntu and Azure Kubernetes Service (AKS), refer to the following topics:

DevOps Test Hub upgrade on Ubuntu

When you want to use the latest version of HCL DevOps Test Hub (Test Hub) on Ubuntu, you must upgrade Test Hub from a previous version.

Before you upgrade Test Hub, you must ensure that you have completed the following tasks:

- Installed Helm. For more information, refer to the [Helm](#) documentation.
- Completed all test executions that are running on the existing version of Test Hub.
- Stopped all stub executions that are running on the existing version of Test Hub.
- Canceled any scheduled test runs that have a future date or time.
- Informed users that Test Hub is offline temporarily during the upgrade process.

For more information about specific versions of software requirements, see [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).

You can upgrade to Test Hub from HCL OneTest™ Server 10.5.4 or 10.5.3 by using the helm update command. If you want to upgrade from versions previous to 10.5.3, you must first upgrade to 10.5.3 or 10.5.4.

In upgrade (by new install) method, depending on the existing version of Test Hub, you can upgrade Test Hub by using any of the following options.

To upgrade Test Hub (10.1.0 or later) to 10.5.3, you must perform these tasks in sequence as listed in the following table:

Tasks	Task description	More information
1	Back up the user data from the previous version of Test Hub.	Backing up DevOps Test Hub data on Ubuntu on page 126
2	Uninstall the previous version of Test Hub including entire Kubernetes environment.	Uninstalling DevOps Test Hub from Ubuntu on page 135
4	Install the latest version of Test Hub.	Installing DevOps Test Hub on Ubuntu on page 109
5	Migrate the user data after the installation is complete.	Migrating user data on Ubuntu from 10.1.2 or earlier on page 122 Or Migrating user data on Ubuntu from 10.1.3 or later on page 123

If your existing version of server software is 10.0.2, Fix Pack 1, or earlier, then you cannot directly upgrade to the latest version of Test Hub due to the steps involved in the migration process. Therefore, you must perform the tasks in sequence as listed in the following table:

Tasks	Task description	More information
1	Upgrade Test Hub 10.1.2.	Installation of DevOps Test Hub on Ubuntu on page 107
2	Back up the user data from Test Hub 10.1.2.	Backing up DevOps Test Hub data on Ubuntu on page 126
3	Uninstall Test Hub 10.1.2 including entire Kubernetes environment.	Uninstalling DevOps Test Hub from Ubuntu on page 135
4	Install the latest version of Test Hub.	Installing DevOps Test Hub on Ubuntu on page 109
5	Migrate the user data after the installation is complete.	Migrating user data on Ubuntu from 10.1.2 or earlier on page 122

Upgrading DevOps Test Hub on Azure Kubernetes Service

When you want to use the latest version of Test Hub on Azure Kubernetes Service (AKS), you must upgrade HCL DevOps Test Hub (Test Hub) from a previous version of Test Hub.

Before you begin

- You must have read and been familiar with additional Helm parameters. See [Additional Helm parameters on page 111](#).
- You must have completed the following tasks:
 - Completed the tasks provided in the Prerequisites section. See [Prerequisites for installing DevOps Test Hub on Azure Kubernetes Service on page 94](#).
 - Uninstalled NGINX Ingress controller by using the following helm command:

```
helm uninstall ingress-nginx -n ingress-nginx
```

For more information refer to the [NGINX Ingress Controller](#) documentation.



Note: To install Test Hub 10.2.2 or earlier on AKS, you installed the NGINX as Ingress controller. From 10.2.3 the Ingress controller has been changed to Emissary-ingress. Therefore, you must uninstall the NGINX Ingress controller and install the Emissary-ingress, see [Installing Emissary-ingress on page 97](#).

- Installed Emissary-ingress.
- Informed users that Test Hub is offline temporarily during the upgrade process.
- Completed all test executions that are running on the existing version of Test Hub.
- Stopped all stub executions that are running on the existing version of Test Hub.
- Canceled any scheduled test runs that have a future date or time.
- Logged in to the AKS cluster.



Note: You can run the az login command to log in to AKS interactively.

- Subscribed to active Azure subscription.



Note: You can run the az account set --subscription {subscription_name} command to set your active subscription.

Optional. You can back up the user data from the earlier version of the product. If the upgrade fails, then you can use that backup file to restore it to the latest version of Test Hub. See [Backing up the DevOps Test Hub data on Azure Kubernetes Service on page 133](#).






Note: For more information about specific versions of software requirements, see [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).


About this task

You can upgrade to Test Hub from HCL OneTest™ Server 10.5.4 or 10.5.3 by using the helm update command. If you want to upgrade from versions previous to 10.5.3, you must first upgrade to 10.5.3 or 10.5.4.

When you want to install the latest version of Test Hub in the same location that contained the 10.2.0 or later, then you can upgrade Test Hub to 10.5.3 by using the helm upgrade command.

The following table lists the variables that you must replace with the actual value in the command.

Variables	Description
{RESOURCE_GROUP}	The name of the resource group that you created during the creation of the AKS cluster.
{azure_cluster}	The name of the Azure cluster that you created during the creation of the AKS cluster.
{azure_container_registry}	The name of the ACR that you created during the AKS setup.
{my-ingress-dns-name}	<p>The Ingress name and DNS zone value that you provided during the setting up of the AKS cluster.</p> <p>For example, if the Ingress name and DNS zone value that you provided are <code>devops, my-org.nonprod</code>, then <code>{my-ingress-dns-name}</code> value must be <code>devops.myorg.nonprod</code>.</p>
{main}	<p>The release name of your choice.</p> <p> Note: The release name must consist of alphanumeric characters that are in lowercase or "-" (hyphen). The release name must also start with an alphabetic character and end with an alphanumeric character. For example, <code>my-org</code> or <code>abc-123</code>.</p>
{HCL_LICENSE_ID}	<p>The ID of the License Server for the initial team space, if you want to set the license for the first time.</p> <p> Important: When you want to upgrade the product from the previous version, you must configure the value of License Server ID from the Team Space License Configuration page when the installation of the server is complete.</p>
{PASSWORD_SEED}	<p>A value of your choice for the password.</p> <p> Important: This password seed is used to create several default passwords for the server. You must store the password seed securely. When you install Test Hub by using the backup of the user data, you can reuse the password seed. You can use</p>

Variables	Description
	 this seed to restore the backed-up files either on the current or later versions of Test Hub.
{ip_address}.nip.io	<p>The IP address that is assigned to Emissary-ingress.</p> <p>You can run the following command to obtain the IP address of Emissary-ingress:</p> <pre>kubectl get ingress -A</pre>



Remember: The `devops-system` is the name of the namespace that you created during the installation of Test Hub. If you created a namespace by using a different value, then you must use that value in place of `devops-system` in all the instances in this procedure.



Tip: You can run the following command to know the value of the variables that you used during the installation of the previous version of Test Hub:

```
helm get values {release_name} -n {namespace}
```

For example, if your name of the release is `{main}` and the name of the namespace is `devops-system`, then you must run the command as follows:

```
helm get values {main} -n devops-system
```

1. Run the following command before you start the upgrade:

```
kubectl delete deployments,statefulsets
-lapp.kubernetes.io/managed-by=Helm,app.kubernetes.io/instance=$HELM_NAME -n $NAMESPACE
```

2. Run the following command:

```
helm pull --untar oci://hclcr.io/ot/hcl-devops --version 11.0.1
cd hcl-devops
```

3. Run the following command to upgrade Test Hub:



Notes:



- If you enabled the autoscaler feature to use a new container only to run test assets and to open the test editor, then you must add `-f hcl-onetest-server/values-dedicated-nodes.yaml` in the following helm install command.
- If you enabled the autoscaler feature to use a new container only to run test assets, then you must add `-f hcl-devops/values-dedicated-nodes.yaml` in the following helm install command.
- When the ingress domain is accessible to untrusted parties, then you must set the `signup` parameter to `false` in the following command.

```
helm update --install $HELM_NAME . -n $NAMESPACE \
  --create-namespace \
  --set global.domain=$INGRESS_DOMAIN \
  -f values-k8s.yaml \
  -f hcl-devops/values-dedicated-nodes.yaml \
  --set global.persistence.rwxStorageClass=azurefile \
  --set imageRegistry=$IMAGE_REGISTRY \
  --set-literal passwordSeed=$PASSWORD_SEED \
  --set signup=true \
  --set hclFlexnetURL=https://hclsoftware.compliance.flexnetoperations.com \
  --set hclFlexnetID=$HCL_LICENSING_ID
```

4. Run the following script from the `hcl-devops/files` directory to verify and test the installed instance:

```
hcl-devops/files/helm-test-diag.sh {main} -n devops-system
```

Results

On the successful upgrade of Test Hub, the output displays the following information:

- Keycloak URL to manage and authenticate users.
- A URL to access the Test Hub UI.

What to do next

- Configure licenses for team spaces. See [Configuring licenses for team spaces on page 181](#).
- Configure Test Hub. See [Configuration of DevOps Test Hub on page 137](#).

Migration of user data

You can find information about the migration limitations and methods when you want to migrate data such as projects, test resources, configuration details of integrated applications, and so on. You can migrate data that is generated in an earlier version of HCL DevOps Test Hub (Test Hub) after you upgrade to or install the latest version, and then use that data in the latest version.

Limitations for user data migration

Before you can decide on the migration method for the version of Test Hub, you must review the limitations that apply to the migrated data.

Table 18. Migration limitations


Migrating from	Migrating to	Limitation	The workaround, if available
10.5.4 or earlier	11.0.0	<p>You cannot migrate the Jira integration or its associated Xray integration to Test Hub 11.0.0 or later, as the method to authenticate the Jira connection is changed in 11.0.0 or later.</p> <p> Note: The links to the Jira or Xray project that are created in earlier versions will be available and will work after Jira and Xray are reconfigured in Test Hub 11.0.0 or later.</p>	You must reconfigure Jira as a change management system on Test Hub 11.0.0 or later. You must also reconfigure the Xray integration, if configured.
10.5.0	10.5.1	You cannot migrate the IBM® Engineering Test Management data that is generated when it is configured as a quality management solution on Test Hub 10.5.0 to Test Hub 10.5.1.	You must reconfigure IBM® Engineering Test Management as a quality management solution in the latest version of Test Hub.
10.5.0 or earlier	10.5.1	If you configured Jira as a change management system on Test Hub, and subsequently changed the name of the project in Jira, then you cannot then configure the changed project name in the CHANGE MANAGEMENT SYSTEM tab of Test Hub 10.5.1.	You must reconfigure Jira as a change management system in the latest version of Test Hub.
10.5.0 or earlier	10.5.1	If you configured IBM® Engineering Test Management as a quality management solution on Test Hub 10.5.0, and subsequently updated or deleted a Project area in Engineering Test Man-	You must reconfigure IBM® Engineering Test Management as a quality management solution in the latest version of Test Hub.

Table 18. Migration limitations (continued)

Migrating from	Migrating to	Limitation	The workaround, if available
		agement, then you cannot perform the following tasks: <ul style="list-style-type: none"> • Delete your project from Test Hub. • Delete the configured Engineering Test Management from the Test Hub project. 	
10.2.3	10.5.0	If you configured IBM® Engineering Test Management on Test Hub 10.2.3, the information about Engineering Test Management is not visible after you migrate Test Hub from 10.2.3 to 10.5.0 or later.	You must reconfigure IBM® Engineering Test Management as a quality management solution in the latest version of Test Hub.
10.2.3	10.5.0	If you configured Jira on Test Hub 10.2.3, the information about Jira is not visible after you migrate Test Hub from 10.2.3 to 10.5.0 or later.	You must reconfigure Jira as a change management system in the latest version of Test Hub.
10.2.3 or earlier	10.5.0	You cannot migrate edited test assets that were created in Test Hub 10.2.3 or earlier, to Test Hub 10.5.0 or later.	You might have to open and edit the test assets in the latest version of Test Hub.
10.2.0	10.2.1 or later	If you edited test assets in your project on Test Hub 10.2.0 and migrated them, then the values displayed in the Last Changed By field might be incorrect for the edited test assets.	

Migration methods

The data that is generated in Test Hub can be migrated and used in a later version of Test Hub with except for the limitations detailed in [Table 18: Migration limitations on page 120](#).

You can refer to the following table to know the details of the migration when you want to upgrade to the latest Test Hub version:

Table 19. Migration method used during an upgrade

OS / Platform	Server version	Upgrade method	Migration method
Ubuntu	10.1.3 or later	Upgrade by new install	When you upgrade Test Hub by uninstalling the existing version, and installing the required version of Test Hub, then you must perform the migration task after the installation of Test Hub is complete. For more information about the migration process, see Migrating user data on Ubuntu from 10.1.3 or later on page 123 or Migrating user data on Ubuntu from 10.1.2 or earlier on page 122 .
	10.1.2 or earlier		
Azure Kubernetes Service	10.2.3 or later	In-place upgrade	When you upgrade Test Hub by using the helm upgrade command, the migration of user data takes place during the upgrade process.

Migrating user data on Ubuntu from 10.1.2 or earlier

When you want to use the data, which was in your earlier version, in a new version of Test Hub, you can migrate those user data.

Before you begin

You must have upgraded Test Hub to a new version.

About this task

The following table lists the variables that you must replace with the actual value in the command.

Variables	Description
{main}	The name of the release that you provided during the installation of Test Hub.
{backup-release}	The Helm release name of Test Hub where you store the backup data.
{target-release}	The Helm release name of Test Hub where you want to restore the data.



Remember: The `devops-system` is the name of the namespace that you created during the installation of Test Hub. If you created a namespace by using a different value, then you must use that value in place of `devops-system` in all the instances in this procedure.

1. Run the following script from the `hcl-onetest-server/files` directory to create a directory that contains metadata related to the Persistent Volume Claims and their Persistent Volumes:

```
migrate.sh create-pvcs -n devops-system {{main}}
```

2. Run the following script from the `hcl-onetest-base` directory to back up the data:

```
sudo backup.sh create-pvc-links -v ~/migration-pvc-links
```

3. Run the following command to stop the cluster and Test Hub:

```
k3s-killall.sh
```

4. Run the following script from the `hcl-onetest-base` directory to restore the backed-up data:

```
sudo backup.sh restore -v ~/migration-pvc-links --release {backup-release}:{target-release}
{backup-file-name}
```

5. Run the following command to restart Kubernetes and to start Test Hub:

```
sudo systemctl start k3s
```

6. Run the following script from the `hcl-onetest-server/files` directory to merge the data into the server:

```
migrate.sh merge-dbs -n devops-system {{main}}
```

7. Run the following command to remove the resources that were created during the migration process:

```
migrate.sh delete-temp-resources -n devops-system {{main}}
```

8. Run the following script from the `hcl-devops/files` directory to verify and test the installed instance:

```
hcl-devops/files/helm-test-diag.sh {main} -n devops-system
```

Results

You have migrated the user data from the earlier version to the new version.

What to do next

You can perform the following tasks:

- Configure licenses for team spaces. See [Configuring licenses for team spaces on page 181](#).
- Configure Test Hub. See [Configuration of DevOps Test Hub on page 137](#).

Migrating user data on Ubuntu from 10.1.3 or later

When you want to use the data, which was in your earlier version, in a new version of Test Hub, you can migrate those user data..

Before you begin

You must have upgraded Test Hub to a new version.

About this task

The following table lists the variables that you must replace with the actual value in the command.

Variables	Description
{main}	The name of the release that you provided during the installation of Test Hub.
{backup-release}	The Helm release name of Test Hub where you store the backup data.
{target-release}	The Helm release name of Test Hub where you want to restore the data.

The `devops-system` is the name of the namespace that is created during the setting up of the k3s environment.

1. Run the following script from the `hcl-onetest-base` directory to create a directory that contains metadata related to the Persistent Volume Claims and their Persistent Volumes:

```
sudo backup.sh create-pvc-links
```

2. Stop the Kubernetes cluster by running the `k3s-killall.sh` command.
3. Restore the backed-up user data by running the following commands:

```
sudo ./backup.sh restore [options] {backup-file-name}
```

You can use the following parameters along with the restore command:

- `--namespace` or `-n`: Use this parameter to restore a specific namespace from the backup file. If you do not mention the namespace, then volumes from all the namespaces in the backup file are restored. You can map one namespace to another namespace by using a colon (:). The syntax is:

```
--namespace <name of the namespace> [:<target-namespace>]
```

- `--release` or `-r`: Use this parameter if the Helm release name of the server to which the backup is being restored is different than the Helm release name of the server where the backup was taken. The syntax is:

```
--release <backup-release>:<target-release>
```

- `--volumes` or `-v`: Use this parameter to specify the directory path of the volumes. The syntax is:

```
--volumes <path-of-the-directory>
```

- `-k` or `--confirm`: Use this parameter to skip the confirmation step.

4. Restart the Kubernetes cluster by running the `systemctl start k3s` command to start Test Hub.
5. Run the following script from the `hcl-onetest-server/files` directory to create all the missing databases:

```
migrate.sh create-missing-dbs {{main}}
```




Note: You can expect the output of the command to contain several errors on objects that already exist. They do not indicate a problem with the execution of the script.

6. Run the following script from the `hcl-devops/files` directory to verify and test the installed instance:

```
hcl-devops/files/helm-test-diag.sh {main} -n devops-system
```

Results

You have migrated the user data from the earlier version to the new version.

What to do next

You can perform the following tasks:

- Configure licenses for team spaces. See [Configuring licenses for team spaces on page 181](#).
- Configure Test Hub. See [Configuration of DevOps Test Hub on page 137](#).

Backup and restoration of the DevOps Test Hub data

You must back up HCL DevOps Test Hub (Test Hub) data before you uninstall the current version of Test Hub or upgrade to the latest version. You must back up the data to avoid data loss or inaccessibility of data.

A backup is the process of creating a copy of the data on your system and storing it elsewhere, generally on secondary storage. You can then use that copy to recover if your original data is lost or becomes inaccessible.

A restore is a process of copying the backed-up data from the secondary storage and restoring it to the original location. You can restore the backed-up data when your original data is lost or becomes inaccessible.

You must back up and restore the data when you perform the following tasks:

- Move the existing environment to a new system.
- Change the name of the release or namespace that you used during the installation of Test Hub.
- Minimize the downtime of Test Hub during disaster recovery.
- Upgrade to the latest version of Test Hub.

To learn more about backup and restoration of Test Hub data on platforms such as Azure Kubernetes Service (AKS) and Ubuntu, see the following topics:

- [Backup and restoration of DevOps Test Hub data on Azure Kubernetes Service on page 130](#)
- [Backup and restoration of DevOps Test Hub data on Ubuntu on page 126](#)

Backup and restoration of DevOps Test Hub data on Ubuntu

To secure the data of HCL DevOps Test Hub (Test Hub) that is installed on the Ubuntu server, you can back up the data. At any point in time, after you install Test Hub, you can restore the backed-up data.

Backing up DevOps Test Hub data on Ubuntu

When you want to upgrade to a new version of Test Hub or to move your existing environments to new systems, you must back up the data.

Before you begin

You must have completed the following tasks:

- Been assigned the same role that was required to install and uninstall Test Hub.
- Been granted with the `sudo` access.
- Installed Test Hub and all of the Test Hub pods are in `Running` state.
- Communicated to the users that Test Hub might be unavailable for some time until the process is complete.

1. Log in to the Ubuntu server and open a terminal.
2. Run the following command to extract the base:

```
helm pull --untar hclsoftware/hcl-onetest-base --version 13.1054.0
```

3. Change to the `hcl-onetest-base` directory.

This directory contains the `backup.sh` script which is required to complete the backup operation.

4. Create a directory that contains metadata related to the Persistent Volume Claims and their Persistent Volumes by running the following command:

```
sudo ./backup.sh create-pvc-links
```

You can use the following optional parameter along with the `create-pvc-links` command:

`--volumes` or `-v`: Use this parameter to specify the directory path of the Volumes.



Remember:



- If you did not provide the volume parameter, by default, the command creates a sub-directory in the same directory where you have the backup script.
- You must ensure that the directory you create must be empty or not exist to avoid script failure.
- If you specify a directory by using the `-v` parameter, then you must use the same parameter value when you restore the data.

5. Run the following command to stop the cluster and Test Hub:

```
k3s-killall.sh
```

6. Run the following command to create a backup of the existing user data:

```
sudo ./backup.sh create [options] <backup-file-name>
```

After you run this command, a backup of the local Persistent Volumes is created. The backup is created as tar archives that is compressed by using gzip (.tar.gz). The **create** command archives the Persistent Volumes into the `<backup-file-name>`.

You can use the following parameters along with the **create** command:

- `--namespace` or `-n`: Use this parameter to back up the Persistent Volumes in the specified namespace. If you do not mention the namespace, then all the Volumes from all the namespaces are included in the backup. The syntax is:

```
--namespace <name of the namespace>
```

- `--volumes` or `-v`: Use this parameter to specify the directory path of the Volumes. The syntax is:

```
--volumes <path-of-the-directory>
```



Remember:

- If you did not provide the volume parameter, by default, the command creates a sub-directory in the same directory where you have the backup script.
- If you specify a directory by using the `-v` parameter, then you must use the same parameter value when you restore the data.

For example,

```
sudo ./backup.sh create --namespace devops-system my-backup.tar.gz
```

This command creates a backup file named `my-backup.tar.gz` that contains all of the Persistent Volumes associated with pods available in the `devops-system` namespace.

7. Run the following command to restart the cluster and Test Hub:

```
sudo systemctl start k3s
```

Results

You have backed up the data of Test Hub.

Restoring DevOps Test Hub data on Ubuntu

You can restore the data that is backed up in Test Hub at any point in time after you install latest version.

Before you begin

You must have completed the following tasks:

- Been assigned the same role that was required to install and uninstall Test Hub.
- Been granted with the *sudo* access.
- Communicated to the users that Test Hub might be unavailable for some time until the process is complete.
- Backed up the data from Test Hub.

About this task

The following procedure is for the restoration of the server data from the current release. If you are upgrading Test Hub, then you can restore the data from the earlier version to the latest version during the installation of Test Hub. For more information, see the **Related information** section.

1. Log in to the Ubuntu server and open a terminal.
2. Run the following command to extract the base:

```
helm pull --untar hclsoftware/hcl-onetest-base --version 13.1054.0
```

3. Change to the `hcl-onetest-base` directory.

This directory contains the `backup.sh` script which is required to complete the restore operation.

4. Create a directory that contains metadata related to the Persistent Volume Claims and their Persistent Volumes by running the following command if you have not done before:

```
sudo ./backup.sh create-pvc-links
```

You can use the following optional parameter along with the **create-pvc-links** command:

`--volumes` or `-v`: Use this parameter to specify the directory path of the Volumes.

For example, `backup.sh create-pvc-links -v hcl-onetest-base/my-pvc-links` creates a directory called `my-pvc-links` that stores the metadata required to restore your user data.



Remember:



- If you did not provide the volume parameter, by default, the command creates a sub-directory in the same directory where you have the backup script.
- You must ensure that the directory you create must be empty or not exist to avoid script failure.

5. Run the following command to stop the cluster and Test Hub:

```
k3s-killall.sh
```

6. Run the following command to restore the backed-up user data:

```
sudo ./backup.sh restore [options] <backup-file-name>
```

The **restore** command overwrites the existing Persistent Volumes with data from the *<backup-file-name>*.

You can use the following parameters along with the restore command:

- **--namespace** or **-n**: Use this parameter to restore a specific namespace from the backup file. If you do not mention the namespace, then volumes from all the namespaces in the backup file are restored.

You can map one namespace to another namespace by using a colon (:). The syntax is:

```
--namespace <name of the namespace> [:<target-namespace>]
```

- **--release** or **-r**: Use this parameter if the Helm release name of the server to which the backup is being restored is different than the Helm release name of the server where the backup was taken. The syntax is:

```
--release <backup-release>:<target-release>
```

- **--volumes** or **-v**: Use this parameter to specify the directory path of the volumes. The syntax is:

```
--volumes <path-of-the-directory>
```

- **-k** or **--confirm**: Use this parameter to skip the confirmation step.

For example,

```
sudo ./backup.sh restore --namespace test-system:devops-system -r rel-1:rel-2 -k my-backup.tar.gz
```

This command restores the volumes that are backed up from the *rel-1* release in the *test-system* namespace to the *rel-2* release in the *devops-system* namespace and skips the confirmation step.

7. Run the following command to restart the cluster and Test Hub:

```
sudo systemctl start k3s
```



Note:



- The cluster and Test Hub might take some time to restart and until that time, you cannot access the Test Hub URL.
- You can run the `kubectl get pods -A` command to verify that the Kubernetes environment is working. After a while, the status of the pods must be `Running` state.

Results

You have restored the data of Test Hub.

Related information

[DevOps Test Hub upgrade on Ubuntu on page 114](#)

Backup and restoration of DevOps Test Hub data on Azure Kubernetes Service

To secure the data in HCL DevOps Test Hub (Test Hub) that is installed on Azure Kubernetes Service (AKS), you can back up the data. At any point in time, after you install Test Hub, you can restore the backed-up data.



Attention: The instructions provided are an example of how an AKS cluster can be configured to use Velero for backing up and restoring the Test Hub data. The procedures to backup and restore the server data might be not valid in all the scenarios.

You must refer to the Velero and Azure Kubernetes Service documentation to identify how you can configure Velero in your AKS cluster based on your existing AKS setup to backup and restore the Test Hub data.

- [Velero Documentation](#)
- [AKS Documentation](#)

Preparing the Azure Kubernetes Service cluster to back up the DevOps Test Hub data

When you want to back up the HCL DevOps Test Hub (Test Hub) data, you must prepare your Azure Kubernetes Service (AKS) cluster to store the backed-up data.

Before you begin

You must have completed the following tasks:

- Created an Azure subscription along with an **Owner** or **Contributor** and **User-Access Administrator** role on the Azure subscription.

You can create a subscription from [Microsoft Azure](#) if you do not have a valid subscription.

- Downloaded the Velero 1.5.3 plugin for Microsoft Azure from the [Velero repository](#).



CAUTION: The instructions provided in this procedure are an example of how an AKS cluster can be configured to use Velero 1.5.3 Microsoft Azure plugin for backing up and restoring the Test Hub data. The procedures to backup and restore the server data might be not valid in all the scenarios.

- Copied the Velero executable file into one of the directories in the **PATH** environment variable.
- Installed Azure Command-Line Interface (CLI). For more information, refer to the [Azure CLI](#) documentation.

About this task

Velero is one of the tools that is available to back up and restore the server data. You must prepare your AKS cluster where you installed Test Hub before you back up the data by using Velero.

1. Run the following command to log in to AKS interactively:

```
az login
```

Result

The command-line interface opens a browser and displays the Log-in page of AKS.

2. Sign in with your AKS account credentials.
3. Run the following command to set your active subscription:

```
az account set --subscription <subscription_name>
```

You must replace `<subscription_name>` with the name of your active Azure subscription.

4. Run the following command to set the location of your AKS cluster:

```
az configure --defaults location=<location_name>
```

For example, `az configure --defaults location=eastus2`



Tip: You can run the following command to get the list of all the available locations:

```
az account list-locations --query "[].{DisplayName:displayName, Name:name}" -o table
```

5. Run the following commands to create a storage account within the AKS cluster for the Velero to store the backup files:

```
AZURE_BACKUP_SUBSCRIPTION_NAME=<subscription_name>
AZURE_BACKUP_SUBSCRIPTION_ID=$(az account list
  --query="[?name=='$AZURE_BACKUP_SUBSCRIPTION_NAME'].id" | [0]) -o tsv)
AZURE_BACKUP_RESOURCE_GROUP=Velero_Backups

az group create -n $AZURE_BACKUP_RESOURCE_GROUP --location <location_name>

AZURE_STORAGE_ACCOUNT_ID="velero$(date -u +%s)"
az storage account create \
  --name $AZURE_STORAGE_ACCOUNT_ID \
  --resource-group $AZURE_BACKUP_RESOURCE_GROUP \
```

```
--sku Standard_GRS \
--encryption-services blob \
--https-only true \
--kind BlobStorage \
--access-tier Hot
```



Note: You must replace `<subscription_name>` and `<location_name>` with the name of your active Azure subscription and location of the AKS cluster.

6. Run the following command to create a container within the storage account to store the backup files:

```
AZURE_BLOB_CONTAINER=velero
az storage container create -n $AZURE_BLOB_CONTAINER --public-access off --account-name
$AZURE_STORAGE_ACCOUNT_ID
```

7. Run the following commands to obtain a key of Azure storage account and to create a file that contains all the relevant environment variables:

```
AZURE_STORAGE_ACCOUNT_ACCESS_KEY=$(az storage account keys list \
--account-name $AZURE_STORAGE_ACCOUNT_ID \
--query "[?keyName == 'key1'].value" -o tsv)

cat << EOF > ./credentials-velero
AZURE_STORAGE_ACCOUNT_ACCESS_KEY=${AZURE_STORAGE_ACCOUNT_ACCESS_KEY}
AZURE_CLOUD_NAME=AzurePublicCloud
EOF
```



Note: There are several ways Velero can authenticate to Azure. This procedure is based on using a storage account access key. The available values for `AZURE_CLOUD_NAME` in the Storage account access key method are `AzurePublicCloud`, `AzureUSGovernmentCloud`, `AzureChinaCloud`, and `AzureGermanCloud`.

8. Run the following command to install Velero with the restic option by using the Azure storage account key:

```
velero install \
--provider azure \
--plugins velero/velero-plugin-for-microsoft-azure:v1.1.0 \
--bucket $AZURE_BLOB_CONTAINER \
--secret-file ./credentials-velero \
--backup-location-config
resourceGroup=$AZURE_BACKUP_RESOURCE_GROUP,storageAccount=$AZURE_STORAGE_ACCOUNT_ID,storageAccountKeyEnvVar=AZURE_STORAGE_ACCOUNT_ACCESS_KEY,subscriptionId=$AZURE_BACKUP_SUBSCRIPTION_ID \
--use-volume-snapshots=false \
--use-restic
```

Results

You have prepared your cluster to back up the server data.

What to do next

You can back up the server data. See [Backing up the DevOps Test Hub data on Azure Kubernetes Service on page 133](#)

Related information[Velero Documentation](#)[Velero plugin for Microsoft Azure documentation](#)[AKS Documentation](#)

Backing up the DevOps Test Hub data on Azure Kubernetes Service

You might want to back up the HCL DevOps Test Hub (Test Hub) data to protect against data corruption or malicious attacks. You might also want to back up the data when you want to upgrade to a new version of Test Hub.

Before you begin

You must have completed the following tasks:

- Been assigned the same role that was required to install and uninstall Test Hub.
- Installed Test Hub and all of the Test Hub pods are in `Running` state.
- Communicated to the users that Test Hub might be unavailable for some time until the process is complete.
- Prepared your cluster to store the backed-up data by using Velero. See [Preparing the Azure Kubernetes Service cluster to back up the DevOps Test Hub data on page 130](#).

About this task

If you plan to back up the data immediately after you prepared your Azure Kubernetes Service (AKS) cluster, then you must go to [Step 3 on page 133](#) to start the backup process.

1. Run the following command to log in to AKS interactively:

```
az login
```

Result

The command-line interface opens a browser and displays the Log-in page of AKS.

2. Sign in with your AKS account credentials.
3. Run the following command to start the backup process of the Test Hub data:

```
velero backup create {backup_file_name} --include-namespaces=test-system
--default-volumes-to-restic
```



Note: You must replace `{backup_file_name}` with the name that you want to provide for the backup file.



Tip: You can run the `velero backup describe {backup_file_name} --details` command to verify the progress of the backup operation.

Results

You have backed up the data of Test Hub on AKS.

What to do next

You can restore the server data. See [Restoring the DevOps Test Hub data on Azure Kubernetes Service on page 134](#).

Related information

[Velero Documentation](#)

Restoring the DevOps Test Hub data on Azure Kubernetes Service

You can restore the HCL DevOps Test Hub (Test Hub) data that is backed up in Test Hub when required.

Before you begin

You must have completed the following tasks:

- Been assigned the same role that was required to install and uninstall Test Hub.
- Communicated to the users that Test Hub might be unavailable for some time until the process is complete.
- Backed up the data from Test Hub.

About this task

If you plan to restore the server data immediately after you backed up the data from Test Hub that you installed on Azure Kubernetes Service (AKS), then you must go to [Step 3 on page 134](#) to start the restoration process.

1. Run the following command to log in to AKS interactively:

```
az login
```

Result

The command-line interface opens a browser and displays the Log-in page of AKS.

2. Sign in with your AKS account credentials.
3. Run the following command to start the restore process of the Test Hub data:

```
velero restore create --from-backup={backup_file_name} --restore-volumes
```



Tip: You can run the `velero restore describe {backup_file_name} --details` command to verify the progress of the restore operation.



Note: You must replace `{backup_file_name}` with the name that you provided during the backing up of the server data.

Results

You have restored the data of Test Hub on AKS.

Related information

[Velero Documentation](#)

Uninstallation of DevOps Test Hub

When you no longer require HCL DevOps Test Hub (Test Hub), you can uninstall Test Hub. You can uninstall Test Hub from the platform on which you installed Test Hub.

Uninstalling DevOps Test Hub from Ubuntu

When you want to install a new version of HCL DevOps Test Hub (Test Hub) or you want to reinstall Test Hub when an ongoing installation fails, you can uninstall Test Hub and its components from the Ubuntu platform.

Before you begin

You must have completed the following tasks:

- Installed Test Hub.
- Closed Test Hub, any open web browsers, and all other applications that are enabled by Test Hub.
- **Optional:** Backed-up data from the previous version of Test Hub.

About this task

You must replace {main} with the name of the release that you provided during the installation of Test Hub in all the instances in this procedure.

The `devops-system` is the name of the namespace that is created during the setting up of the k3s environment.

1. Log in to the Ubuntu server using an SSH session.
2. Run the following command to stop the workload that is running:

```
kubectl delete all,cm,secret -lexecution-marker -n devops-system
```

3. Run the following command to uninstall Test Hub:

```
helm uninstall {main} -n devops-system
```

The PersistentVolumeClaims and PersistentVolumes that were created during the installation are not deleted automatically. If you reinstall Test Hub, the user data is reused unless you specifically delete those volumes.

4. Run the following commands to delete the entire Kubernetes environment, including all user data:

Ubuntu version	Commands to be run
18.04	<pre>cd hcl-onetest-base sudo ./ubuntu-wipe.sh --confirm</pre>

Ubuntu version	Commands to be run
20.04	<pre>cd hcl-onetest-base sudo HOME=\$HOME ./ubuntu-wipe.sh --confirm</pre>

After you run this command, the system returns to the same state as if `k3s-init.sh` was never run.

! **Important:** If you want to migrate data from a previous version to the latest version of the product, then you must ensure to back up the data before you run this command.

Results

You have uninstalled Test Hub from the Ubuntu platform.

Uninstalling DevOps Test Hub from Azure Kubernetes Service

When you want to install a new version of HCL DevOps Test Hub (Test Hub) or you want to reinstall Test Hub when an ongoing installation fails, you can uninstall Test Hub and its components from the Azure Kubernetes Service (AKS) cluster.

Before you begin

You must have completed the following tasks:

- Installed Test Hub.
- Closed Test Hub, any open web browsers, and all other applications that are enabled by Test Hub.
- **Optional:** Backed-up data from the previous version of Test Hub.

1. Run the following command to log in to AKS interactively:

```
az login
```

Result

The command-line interface opens a browser and displays the Log-in page of AKS.

2. Sign in with your AKS account credentials.
3. Run the following command to uninstall Test Hub:

```
helm uninstall {main} -n devops-system
```

The PersistentVolumeClaims and PersistentVolumes that were created during the installation are not deleted automatically. If you reinstall Test Hub, the user data is reused unless you specifically delete those volumes.

4. Run the following command to delete the resource group from the AKS cluster:

```
az group delete -g <resource_group>
```

You must replace `<resource_group>` with the name of the resource group that you created during the creation of the AKS cluster.

Results

You have uninstalled Test Hub from the AKS cluster.

Configuration of DevOps Test Hub

You can find information about the tasks that you must perform after you installed HCL DevOps Test Hub (Test Hub) to configure Test Hub.

The following table describes the tasks that you can perform as an administrator after you installed Test Hub:

Requirement	Tasks	More information
Required	Review the default user management to know how Test Hub manages the users and their authentication.	User administration on page 137
Required	Import Certificate Authority (CA) into a browser to prevent certificate errors when you access the Test Hub UI.	Certificate authority: Importing and extending lists on page 156
Optional	Copy the third-party application Jars to Kubernetes to run API Suites that uses a transport.	Copying third-party application Jars to Kubernetes on page 160
Optional	Configure multi-factor authentication (MFA) for logging into Test Hub.	
Optional	Change the password seed to provide enhanced security to Test Hub.	Changing the password seed on page 162
Optional	Disable the server extensions that are enabled during the installation of Test Hub, if you do not want the server extensions to run on the cluster to save server resources.	Disabling server extensions on page 165

User administration

After you install HCL DevOps Test Hub (Test Hub), you must consider how the software manages users and authentication.

You can review the default user management provided by the server and decide what additional controls you might want to add. If you manage users and authentication through an existing LDAP/AD server, you can review how to use that server to manage Test Hub users.

Enabling the Sign up link on the Login page of DevOps Test Hub

You can enable the **Sign up** link when it is not visible on the **Login** page of HCL DevOps Test Hub (Test Hub) by using the Keycloak UI.

Before you begin

You must have completed the following tasks:

- Installed Test Hub.
- Assigned the role of a Keycloak Administrator.
- Copied the URL of the Keycloak UI, username, and password on completion of the installation of server software.

About this task

The **Sign up** link on the **Login** page of Test Hub is not visible when you install Test Hub by using the default values. The **Sign up** link is hidden to provide the enhanced security to Test Hub. You can enable the **Sign up** link on the **Login** page when you want to provide self-registration of users in Test Hub.

You can enable the **Sign up** link on the **Login** page in two ways:

- You can use the `keycloak.signup.enabled` parameter during the installation of Test Hub.

For more information about installing Test Hub by using the `keycloak.signup.enabled` parameter, see the **Related information** section.

- You can use the Keycloak UI after the installation is complete.

For more information on Keycloak UI, see the **Related information** section.

The following procedure describes steps to be performed to enable the **Sign up** link by using Keycloak UI.

1. Log in to the Keycloak Admin Console.
2. Click **Realm Settings** from the left navigation pane.
3. Click the **Login** tab.
4. Set **User Registration** to **ON**.
5. Click **Save** to apply the changes.

Results

You have enabled the **Sign up** link on the **Login** page of Test Hub. The **Sign up** link is now visible on the **Login** page of Test Hub.

What to do next

You can click the **Sign up** link to create an account in Test Hub.

Related information

[Installing DevOps Test Hub on Azure Kubernetes Service on page 100](#)

[Installing DevOps Test Hub on Ubuntu on page 109](#)

[Keycloak](#)

Default user administration

You installed HCL DevOps Test Hub (Test Hub) and you want to know about how Test Hub manages user access.

Test Hub uses Keycloak (<https://www.keycloak.org/>) to manage and authenticate users.

If you manage and authenticate users by using an LDAP and Active Directory server, you can configure Keycloak to connect to that server. For more information, see [LDAP user administration on page 141](#).

Keycloak uses the concept of a realm to manage and authenticate users. When you install Test Hub, a realm called `testserver` is created for you in Keycloak. All server users belong to this realm and when they log in to Test Hub, they log into that realm.



Important: You must ensure that you have set the realm as `testserver` before you proceed with configuring the LDAP settings.

As an administrator, it is important to consider the following points about the server administration:

- By default, there is no administrator for Test Hub.

Such an administrator is required for accessing additional functions, which includes claiming ownership of projects and unarchiving them. But you can assign administrative privileges to any user. You must do this by adding the admin role to the user in Keycloak.

- You must sign up a user that you want to be the administrator. You must go to the Login page at `https://<fully-qualified-dns-name>:443` and sign up.



Note: Do not use that admin user to perform non-administration tasks. Instead, sign up another user.

- After you sign up the user that you want to be the administrator for Test Hub, you must log in to the Keycloak Admin Console at `https://<fully-qualified-dns-name>:443auth/admin/master/console/#/testserver` to make that user the server administrator.

The default username for the Keycloak administrator is `keycloak`. The password is randomly generated when the software is installed. You can see the password by using the following `kubectl` command:

```
kubectl get secret -n devops-system {{main}}-keycloak-postgresql -o jsonpath="{.data.password}" | base64 --decode; echo
```



Note: You must substitute `{{main}}` with the release name that you used during the installation of Test Hub.

After you log in to the Keycloak Admin Console, from the **Users** page, you can search and select the user that you want to make an administrator. From the **Groups** tab, you can join the user to the **Admins** group.

For more information about assigning user roles, see [Groups](#) in the Keycloak documentation.

Now that you are the server administrator, it is important to consider the following points about the default user management and authentication:

- Minimum password length defaults to 8 characters
- Email verification of new users is turned off
- The Forgot Password feature is turned on by default but no instructions are sent to the user to reset their password
- Forgotten user passwords are changed by you if you do not enable Keycloak to send instructions to reset a password

You can review the following sections about changing the default authentication controls.

Email settings

By default, the `testserver` realm sets the Forgot Password switch on. However, as an administrator, you must enable Keycloak to send an email to the user with instructions to reset their password. If you want to verify an email, you must also enable Keycloak to send an email to the user to verify their email address.

You must provide SMTP server settings for Keycloak to send an email. After you log in to the Keycloak Admin Console, see [Email Settings](#) in the Keycloak documentation.

Then, to set up the email verification, see [Forgot Password](#) in the Keycloak documentation.

Password policy

By default, the `testserver` realm has a password policy where the minimum length of a password is 8. As an administrator, you can update password policies in Keycloak.

After you log in to the Keycloak Admin Console, see [Password Policies](#) in the Keycloak documentation.

User password

If you did not enable Keycloak to send instructions to a user about how to reset a password, you must use the Keycloak Admin Console to change their password for them.

After you log in to the Keycloak Admin Console, see [User Credentials](#) in the Keycloak documentation.

User deletion

When a user is inactive or no longer needs to access the server, you can delete that user.

After you log in to the Keycloak Admin Console, see [Deleting Users](#) in the Keycloak documentation.

LDAP user administration

After you installed HCL DevOps Test Hub (Test Hub), you can use a Lightweight Directory Access Protocol and Active Directory HTTP server to manage users and authentication. Some of the following best practices, can help you to use an LDAP/AD server to manage user access to Test Hub.

Test Hub uses Keycloak (<https://www.keycloak.org/>) to manage and authenticate users.

Existing user databases hold user credentials. Keycloak federates these existing external user databases through the concept of storage providers. By default, Keycloak supports an LDAP and Active Directory storage provider. By adding a storage provider, you can map LDAP user attributes into Keycloak. You can also configure more mappings.

Before you configure Keycloak to use an existing LDAP/AD provider, you might consider the following best practices:

- Set up your LDAP/AD provider as a read-only repository so that Test Hub cannot change it.
- Add and remove users in LDAP/AD and not the Keycloak local user database.
- Import and synchronize your LDAP/AD users to your Keycloak local database.
- Map a login style name, for example, `user1@server.com`, by using the `UserPrincipalName` attribute in LDAP/AD to a `username` in Keycloak. If you want the full name of the user as your login style, use the `cn` attribute in LDAP/AD.



Note: The LDAP/AD username attribute must match the LDAP/AD provider username attribute (**Username LDAP attribute**) in Keycloak for the LDAP/AD provider to connect with Keycloak.

The following sections use these best practices to guide you to set up Keycloak to connect to your LDAP/AD HTTP server.

LDAP provider selection in Keycloak

You can use the Keycloak Admin Console to add an LDAP/AD provider.

You can log in to the Keycloak Admin Console at `https://<fully-qualified-dns-name>auth/admin/master/console/#/testserver`. The default username for the Keycloak administrator is `keycloak`. The password is randomly generated when you installed the Test Hub software. You can see the password by using the following kubectl command:

```
kubectl get secret -n devops-system {{main}}-keycloak-postgresql -o jsonpath="{.data.password}" | base64 --decode; echo
```



Note: You must substitute `{{main}}` with the release name that you used during the installation of Test Hub.

After you log in, you can go to the User Federation page to add your provider.

You must select the **Add Ldap providers** option and complete your LDAP/AD connection parameters.

Required settings for a successful connection to your LDAP/AD provider

In the **Add Ldap provider** page, you must complete the fields or select the required values from the list in the field to ensure a successful HTTP connection to your LDAP/AD provider.

Select or enter details in the fields that are displayed in the following sections on the **Add Ldap provider** page:

Section	Field	Action
General options	Console Display Name	Enter a name for the LDAP server that you can identify when it is displayed in the Keycloak Admin Console. For example, you can enter the name as LDAP.
	Vendor	Select the vendor that is providing the LDAP service from the list of providers. For example, you can select Active Directory from the list of vendors.
Connection and authentication settings	Connection URL	Enter the LDAP/AD URL so that Keycloak can connect to your LDAP/AD user database. For example, <code>ldap://<hostname>.<domain></code> .
	Enable StartTLS	Set to <i>On</i> , if you want to use a secure connection.
	Test connection	Click to test the connection with the LDAP server.
	Bind type	Retain the default value set as <i>simple</i> .
	Bind DN	Enter the user ID of the LDAP/AD user database administrator. Make sure that you test the authentication and confirm that the authentication is successful.
	Bind credentials	Enter the password of the user ID of the LDAP/AD user database administrator.
	Test authentication	Click to test the authentication with the LDAP server database.
LDAP searching and updating	Edit mode	Select the <i>READ_ONLY</i> option from the list. This setting means that your LDAP/AD user database is read-only. No user data defined through the mapping of attributes in Keycloak such as the <code>username</code> is written back from the Keycloak user database to your LDAP/AD user database. This read-only setting also means that a new user cannot sign up from Test Hub as no user data can be written to your LDAP/AD user database.
	Users DN	Enter the directory where the LDAP users are listed, for example, <code>cn=Users,dc=Devops,dc=COM</code> .
	Username LDAP attribute	Retain the default value <code>cn</code> for the <code>username</code> , which is a first name, last name, or you can use <code>userPrincipalName</code> , which is <code>username@domain</code> . You might want a login that matches your company style. For example, you might prefer <code>joetester@mycompany.com</code> instead of Joe Tester.

Section	Field	Action
		If you do make this change the <code>username</code> that you specify now must match the <code>username</code> in the Mapper. Keycloak makes this change for you.
	User object classes	Enter the User Object Classes as <code>User</code> because LDAP user records are found based on a user object class.
	User LDAP filter	Enter the filter so that a subset of all your LDAP users are imported as users by Keycloak. For example, you can set up a <code>Devops</code> user group for your LDAP/AD user database such that only those users are imported to Keycloak. <code>(memberOf=cn=Testers,dc=Devops,dc=COM)</code>

Mappers

Keycloak uses mappers to map the user attributes defined in the Keycloak user model such as **username** and **email** to the corresponding user attributes in the LDAP/AD user database. By default, when you saved your settings and created your LDAP/AD provider, the following mappers were created.

The `username` attribute that you specified in the **Username LDAP attribute** must match the `username` attribute defined in the Keycloak mapper for the LDAP/AD user database to connect with Keycloak.

Because you changed the **Username LDAP attribute** from the default value `cn` to `userPrincipalName`, Keycloak made the same change in the mapper called **username** to match.

User synchronization

You must import all users from your LDAP/AD user database by using the option to **Synchronize all users**. Users are imported based on your saved settings when you set up your LDAP/AD provider.

A successful import is followed by a success message with the number of users imported.

You can view all the LDAP/AD database users that were imported and authenticated from the **Users** page in the Keycloak Admin Console.

Users are listed with Username, Email, Last Name, and First Name. The value of the other attributes is fetched from the LDAP/AD user database by using mappers.



Important: If you get a message of a failed import for an LDAP/AD user, then the failed import might be due to any one of the following reasons:

- The **User logon name** field under the **Account** tab of that LDAP/AD user is blank.
- The value of the **Custom User LDAP Filter** field to import only a subset of all LDAP users does not match with the LDAP set of users.



You can check the log file of the Keycloak pod to understand the reason for the failed import.

You can perform the following steps to view the log file of the Keycloak pod:

1. Run the following command to view the list of all pods:

```
kubectl get pods -n devops-system
```

2. Identify the Keycloak pod `ssocloak-0`.
3. Run the following command to view the log of the Keycloak pod:

```
kubectl logs {{main}}-ssocloak-0 -n devops-system
```



Note: Where, `{{main}}` is the name of the release that is provided during the installation of the server software.

The log file of the Keycloak pod is displayed.

A sample of a log file is as follows:

```
12:49:00,535 ERROR [org.keycloak.storage.ldap.LDAPStorageProviderFactory] (default task-40)
Failed during import user from LDAP: org.keycloak.models.ModelException: User returned from LDAP
has null username!
Check configuration of your LDAP mappings.
Mapped username LDAP attribute: userPrincipalName, user DN:
CN=DefaultAccount,CN=Users,DC=DEVOPS,DC=COM,
attributes from LDAP: {whenChanged=[20190527143130.0Z], whenCreated=[20190527143130.0Z],
cn=[DefaultAccount],
userAccountControl=[66082], pwdLastSet=[0]}
```

Managing account settings

You might want to change your password or other settings for your account.


About this task

As an account owner, you can view your account settings, which include your username, email, and first and last name. You can also change all of these settings except the username. Changing your password is also possible from your account settings.



Note: Account settings cannot be changed if HCL DevOps Test Hub (Test Hub) was configured to use an LDAP/AD provider. For more information about server security, see the related links.


- Edit your account details by following these steps:

1. Click the **User** icon  from the menu bar, and select **Account Settings**.

The **Edit account** page is displayed.

2. Edit your user details and save the changes. You can also reset your changes. If you cancel your changes, you return to the Home page.

- Change your password by following these steps:

1. Click the **User** icon  from the menu bar, and select **Account Settings**.

The **Edit account** page is displayed.

2. Click **Change password**.

The Change password page is displayed.

3. Type your current password followed by your new password

4. Confirm the new password by typing it again and save the changes. You can also reset your changes. If you cancel your changes, you return to the Home page.

Related information

[Default user administration on page 139](#)

Configuration of multi-factor authentication in Keycloak

As a *Server Administrator*, you can configure multi-factor authentication that is provided by Keycloak so that the authenticity of the users is validated when they sign in or log into Test Hub.

Test Hub uses Keycloak for user administration and user authentication is one of its features.

Keycloak mandates users create their login credentials at the time of signing up with Test Hub. To enhance the security of Test Hub and to prevent unauthorized users from logging in when user credentials are compromised, the *Server Administrator* can use the multi-factor authentication that is provided by Keycloak.

Multi-factor authentication

The multi-factor authentication (MFA) feature that is provided by Keycloak includes the following methods to authenticate the users when they sign in or log into Test Hub:

- Use of a password that is created by the user when the user creates an account in Test Hub is the default layer of authentication.
- A one-time password (OTP) that is generated by any of the OTP generators that the users install on their mobile device and register their mobile device with Keycloak. The OTP method is an optional layer of authentication that needs to be configured and enabled in Keycloak by the *Server Administrator*.

As a *Server Administrator*, you can configure the OTP policy as a MFA method in Keycloak. See [Setting up a one-time password policy as MFA in Keycloak on page 147](#).

After you set up the OTP policy in Keycloak, you must enable or activate the OTP policy so that the MFA UI is displayed for users at the time of logging into Test Hub. See [Enabling the one-time password policy as the default MFA action in Keycloak on page 150](#).

At times, if users report that their credentials are compromised or have lost their mobile device, the *Server Administrator* can reset the OTP credential of the user that is saved in Keycloak. Resetting the OTP credential enables the users to register their mobile devices again to use the OTP authentication. See [Resetting the one-time password policy for a user in Keycloak on page 150](#).

Setting up a one-time password policy as MFA in Keycloak

As a *Server Administrator*, you can set up a one-time password (OTP) authentication policy in Keycloak as part of the multi-factor authentication (MFA) to validate the authenticity of the users when they sign in or log into Test Hub.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Server Administrator* of Test Hub. See [Default user administration on page 139](#).
- Installed Test Hub. See [Installation of DevOps Test Hub on page 93](#).
- Configured the licenses for Test Hub. See [HCL license portal on page 178](#).

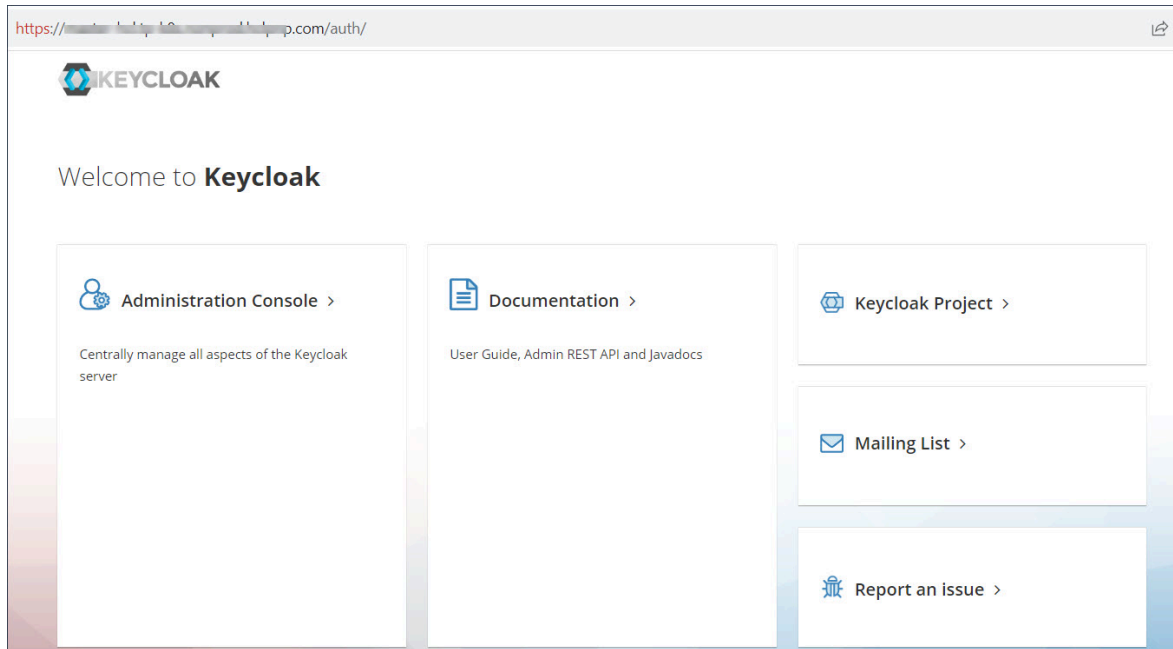
About this task

Keycloak provides an OTP policy that can be enabled and configured in Keycloak for the installed instance of Test Hub. You can refer to the [Keycloak documentation](#) about the OTP policies.

1. Log in to Test Hub, if you are not already logged in.
2. Open Keycloak by entering `/auth` in the **Address bar** after the URL of your server instance, and then press the **Enter** key.

For example, if the URL of the server instance is `https://<myserver.mycompany.com>`, add `/auth` after the URL as in `https://<myserver.mycompany.com>/auth`

The Keycloak dashboard is displayed.



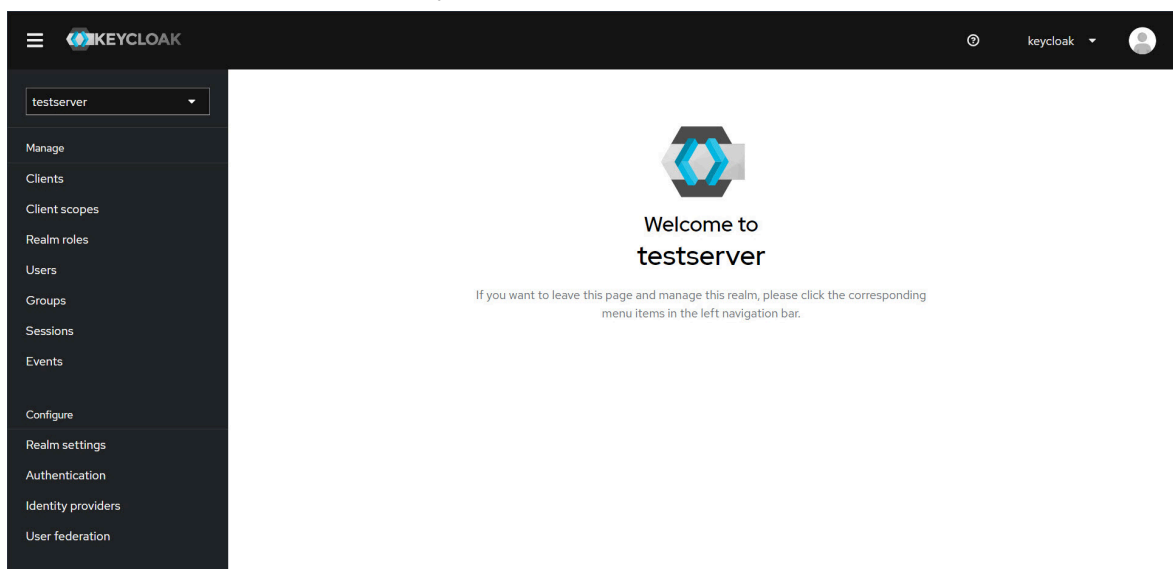
3. Click **Administration Console**.

4. Log into the Keycloak administration console by using the default username as *keycloak* and the password generated for you by Keycloak, if prompted.

The Keycloak console is displayed.

5. Change the realm to *testserver*, if it is not already in this realm.

The *testserver* realm dashboard is displayed.



6. Click **Authentication** in the navigation pane.

The **Authentication** page is displayed.

- Click the **Policies** tab, and then click the **OTP Policy** tab.

The OTP Policy configuration settings are displayed.

Authentication

Authentication is the area where you can configure and manage different credential types. [Learn more](#)

Flows
Required actions
Policies

Password policy
OTP Policy
Webauthn Policy
Webauthn Passwordless Policy

OTP type ? Time based Counter based

OTP hash algorithm ?

Number of digits ? 6 8

Look ahead window ? - +

OTP Token period ? Seconds ▼

Supported applications ?

Reusable token ? Off

Save
Reload

- Read the description provided for each of the fields by clicking ? and decide the values that you want to set for each of the fields, which are different from the default values listed.

You can save the changed configuration or use the default configuration without making any changes.

Results

You configured the settings or retained the default settings for an OTP policy in Keycloak.

What to do next

You must enable the OTP policy in Keycloak as the default option for authentication. See [Enabling the one-time password policy as the default MFA action in Keycloak on page 150](#).

Enabling the one-time password policy as the default MFA action in Keycloak

After a *Server Administrator* sets up a one-time password (OTP) authentication policy in Keycloak, the *Server Administrator* must enable the policy so that Keycloak can deploy the OTP policy as the default multi-factor authentication (MFA) action.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Server Administrator* of Test Hub. See [Default user administration on page 139](#).
- Configured a one-time password policy for MFA in Keycloak. See [Setting up a one-time password policy as MFA in Keycloak on page 147](#).
- Logged into the *testserver* realm in Keycloak.

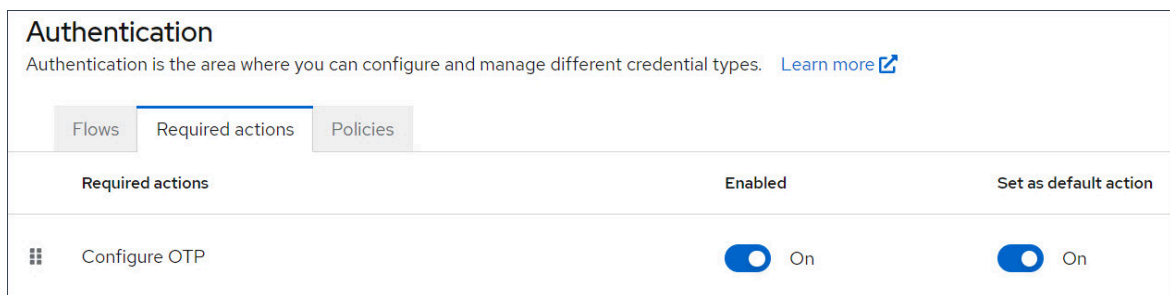
About this task

The usage of an OTP policy for authentication is enabled but not set as the default action in Keycloak. As a *Server Administrator*, you must set the OTP policy as the default MFA action to authenticate users accessing the server instance of Test Hub.

1. Click **Authentication** in the navigation pane.

The **Authentication** page is displayed.

2. Click the **Required actions** tab.
3. Click the toggle under **Set as default action** column for the **Configure OTP** item to change from *Off* to *On*.



Results

You set the OTP policy as the default MFA action.

Resetting the one-time password policy for a user in Keycloak

If multi-factor authentication is enabled for Test Hub, and a user is not able to log in because the mobile device that generates an OTP is lost, the user must request the *Server Administrator* to reset their credentials. The *Server Administrator* must reset the credentials of the user so that the user can register the mobile device again.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Server Administrator* of Test Hub. See [Default user administration on page 139](#).
- Configured a one-time password policy as MFA in Keycloak. See [Setting up a one-time password policy as MFA in Keycloak on page 147](#).
- Enabled a one-time password policy as the default MFA action in Keycloak. See [Enabling the one-time password policy as the default MFA action in Keycloak on page 150](#).
- Received requests from users to enable registration of their new mobile devices for authentication.
- Logged into the *testserver* realm in Keycloak.

About this task

When as a *Server Administrator*, you configure and enable a one-time password (OTP) authentication policy in Keycloak, all users who attempt to log into Test Hub must provide their password and an OTP. Users must install the OTP generators on their mobile devices and register their mobile devices with Keycloak.

If a user loses the mobile device that has the OTP generators installed, then that user cannot provide an OTP to log into Test Hub.


You must then delete the stored credentials of the user for the OTP authentication in Keycloak and re-enable the OTP action for the user. The user must register the new device again for OTP authentication.

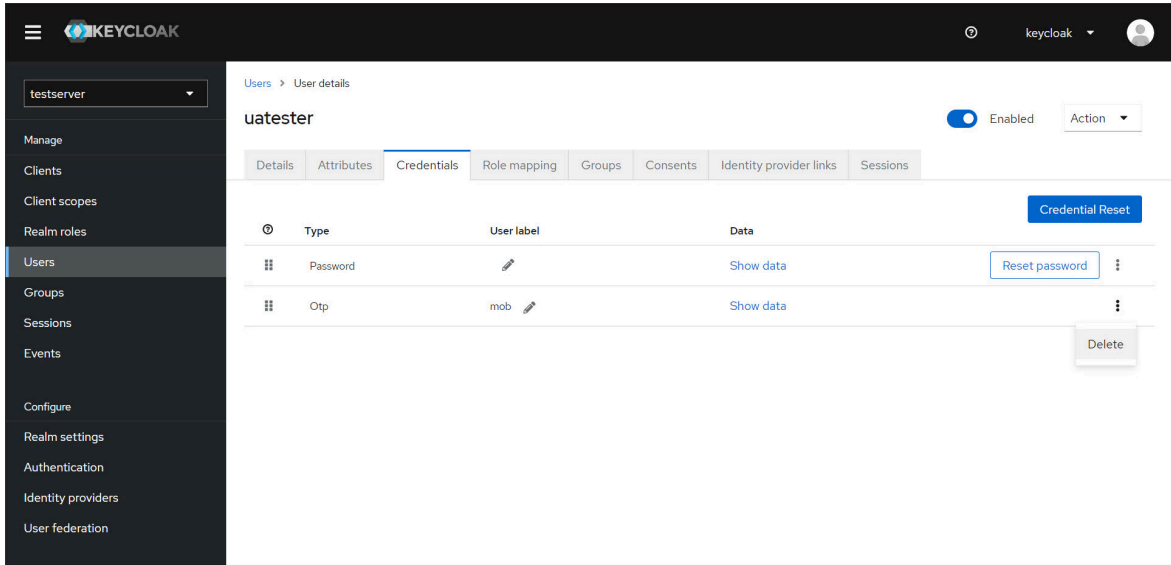
1. Click **Users** in the navigation pane.

The **Users** page is displayed.

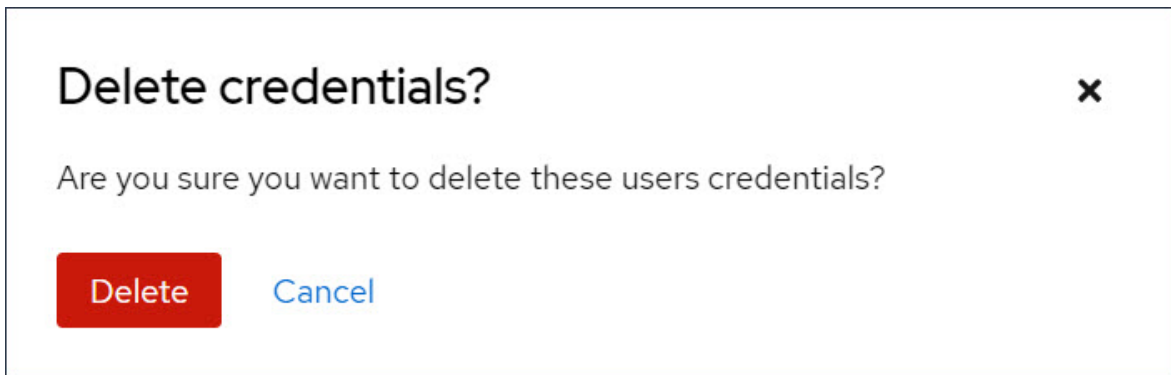
2. Search for the user by entering the username of the user in the **Search** field.
3. Click the username that is displayed as a result of the search.

The **User details** page is displayed.

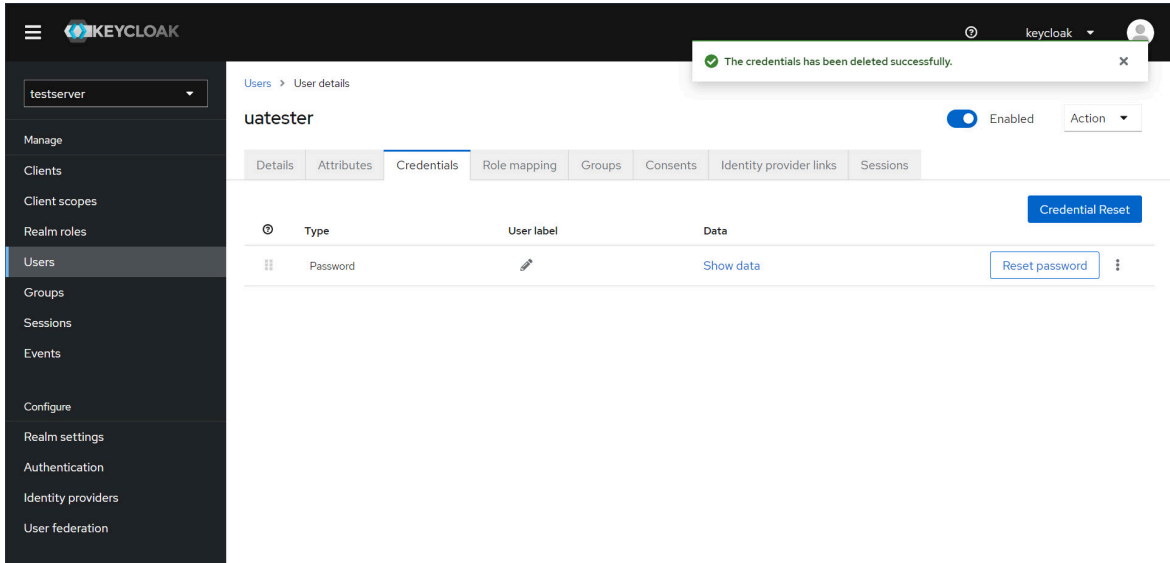
4. Click the **Credentials** tab.
5. Click the menu icon  in the row of **Otp**, and then click **Delete**.



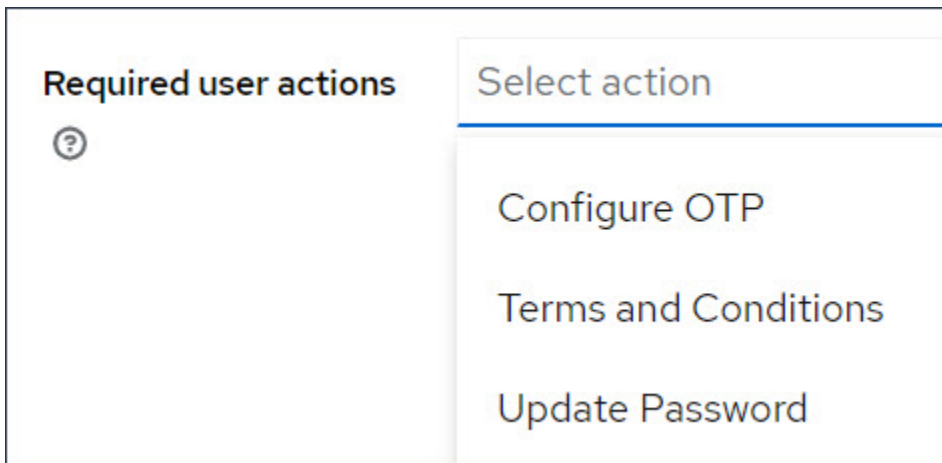
6. Click **Delete** in the **Delete credentials?** dialog.



The saved details of the OTP credential are removed for the user.



7. Click the **Details** tab on the **User** page.
8. Select the **Configure OTP** option from the **Required user actions** list.




9. Click **Save**.


When the user attempts to sign in or log into Test Hub, the **Authenticator Setup** dialog is displayed, and the user can register their new device for the OTP authentication with Keycloak.

English ▾

Authenticator Setup

 You need to set up Authenticator to activate your account.

1. Install one of the following applications on your mobile
 - FreeOTP
 - Google Authenticator
2. Open the application and scan the barcode



[Unable to scan?](#)

3. Enter the one-time code provided by the application and click Submit to finish the setup

Provide a Device Name to help you manage your OTP devices.

Results

You reset the OTP policy for a user.

Managing third-party licenses

When you want to manage your acceptance of any third-party license agreements, you can change them from your **Account Settings** page on HCL DevOps Test Hub (Test Hub).

You can find following information:


- [Accepting third-party licenses on page 155](#)
- [Removing acceptance of third-party licenses on page 156](#)

Accepting third-party licenses

When you want to accept any third-party license agreements if you have not done so already, you can accept them from your **Account Settings** page on HCL DevOps Test Hub (Test Hub).

About this task

When you want to use built-in web browsers such as Chrome or Edge, you must first accept the license agreement and terms of use of these third-party browsers. Test Hub prompts you to accept the license agreements at the time of signing in or logging in. If you do not accept the license agreements at that time, you can accept the license agreements any point in time after you log in to Test Hub.

1. Log in to Test Hub, if you are not already logged in.
2. Click the **User** icon , and then click **Account Settings**.

The **Account Settings** page of your account is displayed.

3. Click **Third Party License Agreements** in the navigation pane.

The **Third Party License Agreements** page is displayed.

The **Google Chrome** and **Microsoft Edge** options are listed with links to the terms of license and service. You can click the links to read about the terms of the license.

4. Click the links to read the terms of the license.

The terms of license and service are displayed in a separate tab or window.

5. Return to the tab or window that displays the **Third Party License Agreements** page in Test Hub.
6. Perform any of the following actions:

- Click the option for the browser that you want to use to accept the license agreement.
- Click both the **Google Chrome** and **Microsoft Edge** options, if you plan to use both the browsers.

7. Click **Save**.


You accepted the license agreements for either the **Google Chrome** or **Microsoft Edge** option or both options.

Results

You accepted the license agreements to use any or both the built-in web browsers.

Removing acceptance of third-party licenses

When you want to remove your acceptance of any third-party license agreements if you have done so already, you can remove your acceptance from your **Account Settings** page on HCL DevOps Test Hub (Test Hub).

1. Log in to Test Hub, if you are not already logged in.
2. Click the **User** icon  and then click **Account Settings**.

The **Account Settings** page of your account is displayed.

3. Click **Third Party License Agreements** in the navigation pane.

The **Third Party License Agreements** page is displayed.

The **Google Chrome** and **Microsoft Edge** options are listed with any or both license agreement options as selected.

4. Perform any of the following actions:
 - Click the option to clear the selection for the browser that you do not want to use.
 - Click both the **Google Chrome** and **Microsoft Edge** options, if you do not want to use both the browsers.
5. Click **Save**.

The options are cleared and your acceptance of license agreement to use these browsers is removed.

Results

You removed your acceptance of the license agreements for any or both browsers.

Certificate authority: Importing and extending lists

When you want browsers and other applications to access HCL DevOps Test Hub (Test Hub), you must enable browsers to trust the certificate authority (CA) of Test Hub. For some applications that integrate with Test Hub, you must also extend the CA lists that are trusted by the applications.

You can find the following tasks to import the certificate authority in the browser that you want to use:

- [Importing Certificate Authority into the Google Chrome browser on page 157](#)
- [Importing Certificate Authority into the Microsoft Edge browser on page 157](#)
- [Importing Certificate Authority into the Mozilla Firefox browser on page 158](#)

You can find the following tasks when you want to extend the CA lists:

- [Extending the trusted CA list on Linux on page 159](#)
- [Extending the trusted CA list on Windows systems on page 159](#)

Importing Certificate Authority into the Google Chrome browser

You must import certificate authority into the Google Chrome browser to prevent certificate errors when accessing the HCL DevOps Test Hub (Test Hub) UI.

Before you begin

You must have completed the following tasks:

- Saved a certificate authority (CA) during the installation of Test Hub.

You can run the following command to get the certificate from the system:

```
kubectl get secret ingress -n devops-system -o jsonpath={.data.ca\\.crt} | base64 -d
```



Remember: *devops-system* is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of the *devops-system*.

- Installed the browser that you want to use to access Test Hub.
1. Open a Google Chrome browser.
 2. Navigate to **Customize and control Google Chrome > Settings**.
 3. Click **Security** from the **Privacy and security** pane.
 4. Click **Manage certificates**, and then **Import**.
 5. Click **Next** in the **Certificate Import Wizard** window.
 6. Click **Browse** and select the CA that you want to import, and then click **Next**.
 7. Select the **Place all certificates in the following store** option to store the CA securely.
 8. Click **Browse**, and then select **Trust Root Certification Authorities** as certificate store.
 9. Click **OK**, and then **Finish**.

Results

You have imported the CA into the Google Chrome browser.

Importing Certificate Authority into the Microsoft Edge browser

You must import Certificate Authority (CA) into the Microsoft Edge browser to prevent certificate errors when accessing the HCL DevOps Test Hub (Test Hub) UI.

Before you begin

You must have completed the following tasks:

- Saved a certificate authority (CA) during the installation of Test Hub.

You can run the following command to get the certificate from the system:

```
kubectl get secret ingress -n devops-system -o jsonpath={.data.ca\\.crt} | base64 -d
```



Remember: *devops-system* is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of the *devops-system*.

- Installed the browser that you want to use to access Test Hub.

1. Type `certmgr.msc` in the **Start** menu **Search** field, and then press **Enter**.
2. Expand **Trusted Root Certification Authorities**, and then select **Certificate**.
3. Right-click on the empty space, and then select **All tasks > Import**.
4. Click **Next** in the **Certificate Import Wizard** window.
5. Click **Browse** and select the CA that you want to import, and then click **Next**.
6. Select the **Place all certificates in the following store** option to store the CA securely.
7. Click **Browse**, and then select **Trust Root Certification Authorities** as certificate store.
8. Click **Next**, and then **Finish** to import the certificate.

Results

You have imported the CA into the Microsoft Edge browser.

Importing Certificate Authority into the Mozilla Firefox browser

You must import Certificate Authority (CA) into the Mozilla Firefox browser to prevent certificate errors when accessing the HCL DevOps Test Hub (Test Hub) UI.

Before you begin

You must have completed the following tasks:

- Saved a certificate authority (CA) during the installation of Test Hub.

You can run the following command to get the certificate from the system:

```
kubectl get secret ingress -n devops-system -o jsonpath={.data.ca\\.crt} | base64 -d
```



Remember: *devops-system* is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of the *devops-system*.

- Installed the browser that you want to use to access Test Hub.

1. Open a Mozilla Firefox browser.
2. Enter `about:preferences#privacy` in the address bar, and then press **Enter**.
3. Locate and click the **View Certificates** option.

4. Select **Authorities** tab, and then click **Import**.
5. Select the CA that you want to import, and then click **Open**.
6. Select **Trust this CA to identify websites** in the **Downloading Certificate** window.
7. Click **OK** to import the CA into the browser, and then close the window.
8. Restart Firefox.

Results

You have imported the CA into the Mozilla Firefox browser.

Extending the trusted CA list on Linux

HCL DevOps Test Hub (Test Hub) trusts the certificates signed by a recognized Certified Authority (CA). When you have any additional CAs, you must extend the trusted certificates list by using an environment variable on Linux.

Before you begin

You must have saved the CA assigned to Test Hub.



Note: You can contact the administrator of Test Hub if you do not have a copy of the CA.

1. Open the terminal.
2. Run the following command to create an environment variable on the Linux operating system:

```
export NODE_EXTRA_CA_CERTS= {path of the downloaded CA with the file extension}
```

For example,

```
export NODE_EXTRA_CA_CERTS=/opt/mycert/ca-master.crt
```

Results

You have extended the trusted CA certificates list on Linux.

Extending the trusted CA list on Windows systems


HCL DevOps Test Hub (Test Hub) trusts the certificates signed by a recognized Certified Authority (CA). When you have any additional CAs, you must extend the trusted certificates list by using an environment variable on Windows systems.

Before you begin

You must have saved the CA assigned to Test Hub.



Note: You can contact the administrator of Test Hub if you do not have a copy of the CA.

1. Right-click the Windows icon  from the Windows taskbar, and then click **System**.
2. Click **Advanced system settings** from the **Related setting** section.

3. Click the **Advanced** tab, and then click **Environment Variables**.
4. Click **New** from the **System variables** section.
5. Enter `NODE_EXTRA_CA_CERTS` in the **Variable name** field.
6. Enter the path of the downloaded CA with the file extension in the **Variable value** field.

For example, if you stored the `ca-master.crt` CA in the `C:\Users\Desktop\cert` location, then enter the **Variable value** field as `C:\Users\Desktop\cert\ca-master.crt`

7. Click **OK**.

Results

You have extended the trusted CA certificates list on Windows systems.

Copying third-party application Jars to Kubernetes

You can run API Suites in a project on HCL DevOps Test Hub (Test Hub). If the API Suite uses a transport and the transport requires third-party application `Jar` files for a successful run, you must ensure that the third-party application `Jar` files are available at the test run time. To achieve this, you must copy the third-party application `Jar` files to the computer where Test Hub is installed on Kubernetes.

Before you begin

If you want to copy the third-party application `Jar` files to the remote Docker host, see [Copying third-party application Jars to a remote Docker host on page 406](#).

You must have server administrator or cluster administrator privileges.

You must have completed the following tasks:

- Identified the third-party application `Jar` files that are required and copied the files. See [Test run considerations for API Suites or API tests on page 373](#).
- Copied the third-party application `Jar` files from HCL DevOps Test Integrations and APIs (Test Integrations and APIs) to the directory or folder in Kubernetes from where you can run the `kubectl` commands.

About this task

You can copy the third-party application `Jar` files to the folder that is specific for the application under the `/data/<application_name>` folder. You need not extract the files.

You can perform this task any time after you have installed Test Hub and you plan to run an API Suite that meets any of the following conditions:

- The API Suite uses transports and the transports require third-party application `Jar` files to be available at the test run time.
- The API Suite in the project has a results database configured.

You can copy the JDK version `Jar` files that you have used to create the JUnit tests.

You can get help on the kubectl commands by running the command: `kubectl cp --help` from the `/kube` directory. For more information, refer to the [kubectl documentation](#).

1. Use the following table to find the name of the folder that corresponds to the specific third-party application for the transport used in the API Suite.

Table 20. Name of the folder for the application

Application	Name of the folder to use
Camel	Camel
CentraSite	CentraSite
CICS	CICS
Coherence	Coherence
Database	JDBC
IMS	IMS
Integra	Integra
JMS	JMS
SAP RFC	SAP
Software AG Universal Messaging	SoftwareAGUM
TIBCO EMS	TIBCO
TIBCO Rendezvous	
TIBCO SmartSockets	
WebSphere Application Server Service Integration Bus (SiBus)	WAS
WebLogic	WebLogicJMX
Software AG webMethods	webMethods

2. Open the command prompt from the `/kube` directory and copy the `Jar` files to the folder that corresponds to the application by using the following command:

```
kubectl cp <compressed_files> devops-system/{{main}}-userlibs-0:/data/<application_name>/
```

For example, if you are copying the Database `Jar` files, then the name of the folder is `JDBC` and the command to use is:

```
kubectl cp mysql-connector-java.jar devops-system/{{main}}-userlibs-0:/data/JDBC/
```



Note: You must substitute `{main}` with the release name that you used during the installation of Test Hub.

3. Verify whether the `Jar` files are copied by running the following command:

```
kubectl exec {{main}} -userlibs-0 -n devops-system -- ls /data/<folder_created>
```

For example, if you copied the Database `Jar` files, `mysql-connector-java.jar` to the folder `JDBC`, then the command to verify is:

```
kubectl exec {{main}} -userlibs-0 -n devops-system -- ls /data/JDBC
```

The following information is displayed for the `JDBC` folder:

```
mysql-connector-java.jar
```



Note: You must substitute `{main}` with the release name that you used during the installation of Test Hub.

Results

You have successfully copied the third-party application `Jar` files to the folder in Kubernetes.

What to do next

You can configure the API Suite run. See [Configuring an API Suite or an API test run on page 486](#).

Changing the password seed

As an administrator, you can change the password seed that is used when you install HCL DevOps Test Hub (Test Hub) to provide enhanced security to Test Hub.

Before you begin

You must have completed the following tasks:

- Installed Test Hub. See [Installation of DevOps Test Hub on page 93](#).
- Installed the JSON command-line tool, `jq`, and ensured that the `jq` is in your environment `PATH`. For information about `jq`, refer to [jq documentation](#).
- Installed the Curl command line tool. For more information refer to [curl documentation](#).


About this task

When you install Test Hub, you supply a password seed when you run the `helm install` command. This password seed is used to generate several Kubernetes secrets. Kubernetes Secrets can contain the following information:

- The authentication credentials for micro-services.
- An encryption key for the user-created secrets.

When you change the password seed for Test Hub, you must consider the following scenarios:

- Test Hub cannot communicate until you reconcile the passwords which are in Kubernetes Secrets by using the old and a new password seed.
- Users cannot read secret collections or other secrets that they have created in Test Hub until you re-encrypt them using a new password seed.

 **Important:** You must provide an offline token and old password seed that you used during the installation of Test Hub to re-encrypt user secrets.

1. Run the following command to change the password seed for Test Hub:

```
helm upgrade {{main}} ./hcl-onetest-server -n devops-system \
--reuse-values \
--set global.hclOneTestPasswordAutoGenSeed= {my-new-super-secret}
```

 **Notes:** You must substitute the value of the following variables with the actual value in the command:

- `{{main}}` with the release name that you used during the installation of Test Hub.
- `{my-new-super-secret}` with a new value of your choice as the password seed.
- You must run the following `helm upgrade` command from the same directory where the `helm install` command was run during the installation of Test Hub. Because the upgrade is dependent on the helm charts and `.yaml` file values used during the run time of the `helm install` command.

2. Run the following script to generate new server secrets from the updated password seed and to save them to the persistent storage:

```
./hcl-onetest-server/files/reconcile-secrets.sh -n devops-system {{main}}
```

3. Run the following command to restart all the pods:

```
kubectrl delete pods -n devops-system \
-lapp.kubernetes.io/instance={{main}} \
-lapp.kubernetes.io/managed-by=Helm
```

4. Run the following commands to re-encrypt the user-created secrets by providing the old password seed:

```
export ACCESS_TOKEN=$(curl -k -X POST {SERVER_URL}/rest/tokens/ \
-H "Content-Type: application/x-www-form-urlencoded" \
-H "accept: application/json" \
-d "refresh_token={OFFLINE_TOKEN}" | jq -r '.access_token')

curl -k -X POST {SERVER_URL}/rest/secrets/re-encrypt/ \
```

```
-H "Authorization: Bearer $ACCESS_TOKEN" \
-H "Content-Type: application/json" \
-d "{\"type\":\"helm\",\"password_auto_gen_seed\":\"{OLD_SEED}\"}"
```



Note: You must substitute the value of the following variables with the actual value in the following commands:

- `{SERVER_URL}` with the URL of your Test Hub UI.
- `{OFFLINE_TOKEN}` with the offline token that belongs to a user with the administrator role.
- `{OLD_SEED}` with the previous password seed that you used during the installation of Test Hub.

5. Run the following command to display the log file of the gateway pod:

```
kubectl logs {{main}} -gateway-abcdefghijklm-abcde -n devops-system
```



Note: You must substitute the value of the following variables with the actual value in the command:

- `{{main}}` with the release name that you used during the installation of Test Hub.
- `abcdefghijklm-abcde` with an identifier that is assigned to the name of the gateway pod.

You can run the `kubectl get pods -n devops-system` command to obtain the identifier that is assigned to the gateway pod.

Result

The following message is displayed when re-encryption is completed:

```
reEncrypt complete. StringyReEncryptor [total=100, fixed=100, broken=0, noop=0]
```

Results

You have successfully changed the password seed for Test Hub.

Server extensions

You can run specific tests in HCL DevOps Test Hub (Test Hub) by enabling the server extensions.

The server extensions are enabled at the time of installation of Test Hub. When the server extensions are enabled, you can perform the following tasks on Test Hub:

- Add repositories that contain the test assets for the HCL AppScan CodeSweep tests, JMeter tests, JUnit tests or Postman collections.
- Configure a test run for the following test types:

- **HCL AppScan CodeSweep**
- **JMeter**
- **JUnit**
- **Postman**

To learn more about enabling and disabling the server extensions, refer to the following topics:

Disabling server extensions

When you do not want the server extensions in HCL DevOps Test Hub (Test Hub) to run on the cluster to save server resources, you can disable server extensions.

Before you begin

You must ensure that the server extensions are enabled at the time of server installation by running the following command:

```
kubectl get pods -n devops-system
```

The status of the server extensions that are enabled is displayed as `Running`.

About this task

You must substitute `{{main}}` with the release name that you provided during the installation of Test Hub in the following procedure.

The `devops-system` is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of `devops-system` in all the instances in this procedure.

1. Log in to the platform that you used to install Test Hub.

For example, you can log in to the Ubuntu server using an SSH session if you installed Test Hub on Ubuntu.

2. Run the following commands that depend on the server extension you want to disable:

To disable the server extension	Run the command
CodeSweep	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestCodesweepEnabled=false</pre>
JMeter	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestJMeterEnabled=false</pre>

To disable the server extension	Run the command
JUnit	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestJUnitEnabled=false</pre>
Postman	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestPostmanEnabled=false</pre>
All server extensions	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestPostmanEnabled=false \ --set global.hclOneTestJMeterEnabled=false \ --set global.hclOneTestJUnitEnabled=false \ --set global.hclOneTestCodesweepEnabled=false</pre>

3. Run the following command to verify the status of the extensions:

```
kubectl get pods -n devops-system
```



Note: The pod for the server extension that you disabled is not displayed.

Results

You have disabled the required server extensions on Test Hub.

Enabling server extensions

If you find that the server extension is disabled in HCL DevOps Test Hub (Test Hub) and you want to run the tests supported by the server extension, then you must enable them.

Before you begin

You must ensure that the server extensions are enabled at the time of server installation by running the following command:

```
kubectl get pods -n devops-system
```

The status of the server extensions that are enabled is displayed as `Running`. If the server extension you want to use is not displayed as running, then you must enable the server extension.

About this task

You must substitute `{{main}}` with the release name that you provided during the installation of Test Hub in the following procedure.

The `devops-system` is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of `devops-system` in all the instances in this procedure.

1. Log in to the platform that you used to install Test Hub.

For example, you can log in to the Ubuntu server using an SSH session if you installed Test Hub on Ubuntu.

2. Run the following commands that depend on the server extension you want to enable:

To enable the server extension	Run the command
CodeSweep	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestCodesweepEnabled=true</pre>
JMeter	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestJMeterEnabled=true</pre>
JUnit	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestJUnitEnabled=true</pre>
Postman	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestPostmanEnabled=true</pre>
All server extensions	<pre>helm upgrade {{main}} -n devops-system ./hcl-onetest-server \ --reuse-values \ --set global.hclOneTestPostmanEnabled=true \ --set global.hclOneTestJMeterEnabled=true \ --set global.hclOneTestJUnitEnabled=true \ --set global.hclOneTestCodesweepEnabled=true</pre>

3. Run the following command to verify a specific extension or all the extensions are enabled:

```
kubect1 get pods -n devops-system
```

Result

The status of the specific extension or all the extensions in the pod is displayed as `Running`.

Results

You have enabled the required server extensions on Test Hub.

Related information

[Configuring a run of a JMeter test on page 665](#)

[Configuring a run of a JUnit test on page 674](#)

[Configuring a run of a Selenium test on page 704](#)

[Configuring a run of a Postman test on page 683](#)

[Configuration of a scan of an application code on page 716](#)

Team space administration

You can find information about team spaces including the initial team space on HCL DevOps Test Hub (Test Hub) along with the tasks that you can perform to administer a team space.

You can find the following information about team spaces:

- [Team space overview on page 168](#)
- [Creating a team space on page 171](#)
- [Tasks in a team space on page 173](#)
- [Tasks in projects in a team space on page 177](#)
- [Management of licenses in DevOps Test Hub on page 179](#)

Team space overview

Team spaces are logical partitions of HCL DevOps Test Hub (Test Hub). Team space is an exclusive, separate, and secure space on Test Hub that enables you to share a single installation among multiple teams.

You can find the following information about team spaces:

- [Initial team space on page 168](#)
- [Team spaces on page 169](#)
- [Migration from earlier releases of Test Hub on page 170](#)

Initial team space

After you install Test Hub and log in for the first time, you enter the **Initial team Space**.

The initial team space is the default logical partition that is created on an instance of Test Hub.

As a server administrator, you must consider the following points:

- After you install Test Hub and log in for the first time, the **Projects** page of the initial team space is displayed.
- You must configure the license server for the initial team space.
- Add or remove users in the team space and assign them different roles in a team space.
- You can assign the role of a *Team Space Owner* to any other member in the initial team space.
- Change the type of the initial team space from the default type of *permissive*.

The following table describes the type that you can assign to a team space:

Type	Description
Permissive	Specifies that any licensed user of Test Hub can access a team space and get assigned the default roles of a <i>Project Owner</i> and <i>Member</i> in the team space.
Restrictive	Specifies that all licensed users of Test Hub must request for access to a team space and must be provided access and assigned roles by the <i>Team Space Owner</i> . The Restrictive type of team space can be either Public or Private . A Public team space is visible in the team spaces dashboard, while a Private team space is not visible in the team spaces dashboard.

- You can evaluate the requirement and create team spaces other than the initial team space.

As a licensed user, you must consider the following points:

- When you log in for the first time immediately after the installation of Test Hub, the **Projects** page of the initial team space is displayed.
- After you log in to the initial team space, you are assigned the default roles of *Project Creator* and *Member* of the initial team space.
- On subsequent logins, the **My Projects** page in the initial team space is displayed as the default page.

Team spaces

You can either work with the initial team space or you can create multiple team spaces on Test Hub. The team space facilitates users of the same team space to work together.

Apart from being a container for projects, a team space facilitates the registration of remote Docker hosts and Resource Monitoring agents, designing the system model, and viewing the registered agents, Dockers, or intercepts.

After you create a team space as a **server administrator**, you must consider the following actions:



Note: You are assigned the default role of a *Team Space Owner* for team spaces that you create.

- Add users in the team space and assign them different roles in a team space.
- Assign the role of a *Team Space Owner* to other team space members, if you want other members to own the team space.



Note: As a *Team Space Owner*, you can remove any member of your team space from the assigned role. The member who is removed from a team space, can send a request and become a member of that team space after the request is accepted.

- If the *Team Space Owner* is no longer a member of a team space, then you as the server administrator can access that team space with the default role of a *Team Space Owner*. You can then assign the role of a *Team Space Owner* to any other member.
- Delete the initial team space or any other team space that you created, if required.



Note: If you delete the initial team space, the users of the initial team space are redirected to the **Team Space Dashboard** page.

You as a server administrator or the *Team Space Owner* must perform the following tasks in a team space before other members can start using the team space:

- Configure the license server.
- Configure the team space type as either *Restrictive* or *Permissive*.
- Configure the SMTP server and add members who can receive notifications about the operations performed in the team space.

As a member of a team space, you must consider the following points:

- When you log in for the first time immediately after the installation of Test Hub, the **Projects** page of the team space is displayed.
- After you log in to the team space, you are assigned the default roles of a *Project Creator*, and *Member* of the team space.
- You can request to become a member of any other team space. After your request is approved, you are assigned the default role of a *Member* of that team space.
- You can become a member of either a **Permissive** or **Restrictive** type of team space.
- On subsequent logins, the **Team Space Dashboard** page is displayed as the default page.

Migration from earlier releases of Test Hub

The initial team space is the only logical partition in Test Hub 10.1.3. If you migrate from an earlier release of Test Hub to 10.2 or later, then all the assets and resources that you created and associated with the initial team space are migrated to associate with the initial team space in Test Hub 10.2 or later.



Note:

The results of migration are as follows:

- All projects are contained in the initial team space.
- If you created a repository in a team space, then you can find the repository in the initial team space in 10.2.
- All Dockers, agents, and intercepts that you registered with the server are then registered with the initial team space and you can view from the **Agents and Intercepts** page of the initial team space.
- If you configured a resource monitoring agent in any project, then the details of the configured agent are available in the project.
- You can find the details of SMTP configuration on the server. Only you as a server administrator can view the details of SMTP configuration.
- The license server that is configured for the server for earlier release of Test Hub is then configured for the initial team space.
- All users are assigned the role of a *Project Creator* and *Architect* of the initial team space.
- All users with server administrator roles are assigned the default role of a *Team Space Owner* of the initial team space.

**Notes:**

- All users of earlier releases are migrated to Test Hub 10.2 or later. You can view the list of all migrated users in the application only when they log into the application at least once.
- After you migrate the server data from a previous version to a newer version, the **Project Owner** must log in to Test Hub so that other members of the project can access the test assets in that project.

Related information

[Team space administration on page 168](#)

[Default user administration on page 139](#)

Creating a team space

To manage projects in the logical containers of HCL DevOps Test Hub (Test Hub), you must create team spaces.

Before you begin

You must have been assigned the role of a server administrator of Test Hub.


About this task

As a server administrator, you can create multiple team spaces on Test Hub.

1. Log in to Test Hub.

Result

The **Projects** page of the initial team space is displayed.

2. Click the **Settings** icon  from the left navigation pane, and then click **Create team space**.

Result

The **New Team Space** dialog is displayed.

3. Provide a name to the team space.

Result

The **URL alias** field is automatically filled.



Note: If you modify the name of a team space, the **URL alias** is also modified automatically.

4. **Optional:** Modify the **URL alias** for the team space, if required.



Note: You must enter the **URL alias** that starts with a letter.

5. Select an option to set the permissions to access the team space. You can select any one of the following options:

- **Permissive** – When a team space is set as **Permissive**, the new user gets the permissions automatically on sending the request to become a member of that team space.
- **Restrictive** – When a team space is set as **Restrictive**, the *Team Space Owner* provides permission to the new user who becomes a member of that team space.

You can set the visibility mode of the **Restrictive** type team space as **Public** or **Private**.

- The **Public** team space is visible to all users in the **Team Space Dashboard**.
- The **Private** team space is visible only to the members of that team space in the **Team Space Dashboard**.



Note: If you select the **Restrictive** type team space, then the default setting of the team space becomes **Public**.

6. Provide a brief description of the team space.
7. Select the owner of the team space from the list of users who accessed the application.

The selected user is assigned the role of a *Team Space Owner*.



Note: You can assign the *Team Space Owner* role to only one user when you create a team space.

8. Click **Create**.

Result

The dashboard of the team space is displayed.

Results

You have created a team space.

If you are assigned the role of a *Team Space Owner*, then you can view the name of the team space that you created in the navigation pane.

If you assigned any other licensed user with the role of a *Team Space Owner*, then you can view the team space under the **Other Team Spaces** panel.

What to do next

You must configure the license for your team space. See [Configuring licenses for team spaces on page 181](#).

Tasks in a team space

You can find information about the tasks that you can perform as a *Member*, *Project Owner*, *Architect*, or *Team Space Owner* in a team space on HCL DevOps Test Hub (Test Hub).

The roles assigned in a team space for users are as follows:

Persona	Initial team space		Any team space	
	Role	Notes	Role	Notes
Server administrator	<i>Team Space Owner</i>	Default role assigned	<i>Team Space Owner</i>	Default role assigned
Licensed user	<i>Team Space Owner</i>	Role assigned by Server administrator	<i>Team Space Owner</i>	Role assigned by Server administrator
	<i>Architect</i>	Role assigned by <i>Team Space Owner</i>	<i>Architect</i>	Role assigned by <i>Team Space Owner</i>
	<i>Project Creator</i>	Default role assigned	<i>Project Creator</i>	<ul style="list-style-type: none"> • Default role assigned in a <i>Permissive</i> team space. • Role assigned by <i>Team Space Owner</i> in a <i>Restrictive</i> team space.
	<i>Member</i>	Default role assigned	<i>Member</i>	Default role assigned in either a <i>Permissive</i> or <i>Restrictive</i> team space.


The following table shows the tasks that can be performed by licensed users with their assigned roles in a team space:

Tasks	Team Space Owner	Architect	Project Creator	Member
Configuring licenses	✓			

Tasks	Team Space Owner	Architect	Project Creator	Member
Configuring a Simple Mail Transfer Protocol (SMTP) server to send email notifications.	✓			
Changing the license server configuration	✓			
Adding members in a team space	✓			
Assigning different roles to other team space members	✓			
Modifying configuration details of a team space	✓			
Deleting a team space	✓			
Renaming a team space	✓			
Adding or removing the team space repository from a team space	✓	✓		
Designing a system model	✓	✓		
Creating projects in a team space	✓		✓	
Adding, removing, or modifying remote Docker hosts	✓	✓	✓	✓
Adding agents	✓	✓	✓	✓
Adding, modifying, or removing resource monitoring sources	✓			
Connecting resource monitoring agents	✓			
Adding resource monitoring labels	✓			
Working with projects in a team space	✓	✓	✓	✓

The following table provides the links to the information about the tasks that can be performed by users with different roles in a team space:

Tasks	Roles	More information
Configuring a team space license server	<i>Team Space Owner</i>	Configuring licenses for team spaces on page 181
Configuring the Named Users	<i>Team Space Owner</i>	Adding users to the Named Users list on page 184

Tasks	Roles	More information
Modifying the configuration of a team space	<i>Team Space Owner</i>	Modifying the configuration of a team space on page 1001
Viewing the configuration of a team space	All members of a team space	Viewing the configuration of a team space on page 1003
Reassigning the role of team space members	<i>Team Space Owner</i>	Managing members and their roles in a team space on page 1006
Becoming a team space member	<ul style="list-style-type: none"> • <i>Team Space Owner</i> • All users can send a request 	Becoming a team space member on page 1005
Viewing team spaces	All members of a team space  Notes: <ul style="list-style-type: none"> • The Permissive and public Restrictive team spaces are visible to all the users. • The private Restrictive team spaces are only visible to its members. 	Viewing team spaces on page 997
Deleting a team space	<i>Team Space Owner</i>	Deleting a team space on page 1009
Adding a project in a team space	<i>Project Creator</i>	Adding a project on page 1013
Viewing existing projects	All members of a team space	Viewing active projects on page 1018
Viewing details about the users or license server	All members of a team space	Viewing license details of the team space on page 186
Installing remote agents and configuring them to register with Test Hub	All members of a team space	Management of agents on page 386
Registering a remote Docker host with Test Hub	All members of a team space	Registering a remote Docker host on page 408
Editing configurations of a remote Docker host	All members of a team space	Editing configurations of a remote Docker host on page 412

Tasks	Roles	More information
Unregistering a remote Docker host	All members of a team space	Unregistering a remote Docker host from DevOps Test Hub on page 416
Viewing remote agents, remote Docker hosts, or intercepts that are registered with Test Hub in the initial team space	All members of a team space	<ul style="list-style-type: none"> • Viewing remote Docker hosts that are registered with DevOps Test Hub on page 410 • Viewing performance agents that are registered with DevOps Test Hub on page 388 • Viewing API agents that are registered with DevOps Test Hub on page 395 • Viewing intercepts that are registered with a team space on DevOps Test Hub on page 746
Adding resource monitoring sources and agents	<i>Team Space Owner</i>	Resource monitoring on page 845
Adding resource monitoring labels to the sources	<i>Team Space Owner</i>	Controlling resource monitoring sources in a schedule on page 878
Viewing the performance metrics	All members of a team space	Viewing resource monitoring usage data on page 885
Creating a model of the system under test	<i>Architect</i>	Creating a system model on page 347
Adding a repository to a team space	<ul style="list-style-type: none"> • <i>Team Space Owner</i> • <i>Architect</i> 	Adding a repository to a team space on page 345
Viewing an existing system model	All members of a team space	Viewing the system model on page 369

Related information

[Team space administration on page 168](#)

Tasks in projects in a team space


You can find information about the roles that you can assign in a project for members of a team space and the tasks that you can perform in projects in a team space.

Members of a team space with the role of a *Project Creator* can create projects in the team space and become the *Owner* of a project. The owners of projects can add *Members* of the team space to a project and assign them the following roles:

- *Owner*
- *Tester*
- *Viewer*

For the list of actions that members assigned any of the roles can perform, see [Managing access to the server projects on page 1021](#).

All projects are created in a team space. When you open a project, the options that are displayed in the navigation pane are specific to the projects within a team space.

When you are working on a project in a team space and you want to return to the options available to work in the team space, you can click the **Home** icon  in the navigation pane.



Note: You must be assigned the role of an *Owner*, a *Tester*, or a *Viewer* of a project.

The following table lists the tasks that members of a team space can perform while working with projects in a team space:

Tasks	More information...
Viewing projects in a team space.	Viewing active projects on page 1018
Adding a project.	Adding a project on page 1013
Adding a repository that contains test assets and resources that you want to run on the server.	Adding repositories to a server project on page 1014
Adding secrets to a classification.	Secrets configuration on page 1017
Encrypting the data set resources.	Managing an encrypted dataset on page 243
Configuring a change management system.	Configuration of a change management system on page 891
Adding members to a project and configuring their roles.	Adding users to a project on page 1017
Selecting the global branch in a project to view the test assets and resources.	Selecting the global branch in a project on page 1033

Tasks	More information...
Adding agents to your project, if you are using remote agents as a location to run certain tests.	Adding a performance agent to a project on page 389
Adding remote Docker, if you are using a remote Docker host as a location to run tests.	Adding a remote Docker host to the project for running tests on page 411
Reading the considerations that you must take into account before you configure a test run.	Prerequisites to running tests on page 371
Reading the considerations that you must take into account before you configure a run or start a virtual service.	Prerequisites for running HTTP virtual services on page 742
Configuring a run of the test assets.	Test run configurations on page 446
Configuring a run of the virtual service resource.	Configuring a run of a virtual service on page 750
Viewing the progress of a test run.	Viewing the progress of running test assets on page 731
Managing a running test.	Management of running tests on page 727
Viewing the results of a test run after the run is completed.	Test results on page 788
Reviewing the test results and creating defects for test runs from the Results page.	Creating Jira issues for the test results on page 895

HCL license portal

Licensing for HCL DevOps Test Hub (Test Hub) is administered through [HCLSoftware License & Download portal](#). This portal is a FlexNet-based web application that helps to manage software entitlements and licenses.

When a software order is placed and acknowledged, a software entitlement is created. You must then follow the instructions in the Software Order Acknowledgment document that you receive to activate your entitlement, create devices, and download the software from the portal.

The license portal provides both software distribution and management of your software entitlements that are purchased from HCLSoftware. The portal provides control and flexibility on how to consume your licenses. An organization identifies one of its resources as a License Manager (also called Tech or Portal Admin) who is familiar with the language of licenses.

For more information about the [HCLSoftware License & Download portal](#), you can refer to the following knowledge articles:

- [What is the HCLSoftware License & Download portal \(FlexNet portal\)?](#)
- [How to find HCL Product Releases in HCLSoftware License & Download portal](#)
- [Managing Users on the HCLSoftware License & Download portal](#)

Management of licenses in DevOps Test Hub

As a *Team space Owner* or *Server Administrator*, you can configure and manage licenses for your team spaces in HCL DevOps Test Hub (Test Hub). As a member of any team space, you can only view the number of licenses that are configured for the team space. If you are not a member of any team space, you cannot view or perform any operations in any of the team spaces on Test Hub.

You can get all the information of configured license server for different team spaces from various *Team Space Owners*. You can navigate to the Test Hub Swagger UI by accessing the following URL:

```
https://myorg.com/#/Initial/admin/docs
```

Where `myorg.com` is the hostname of Test Hub.

You can then open the **Licensing management** tab, and then click the **GET /rest/licensing/inventory** option. To generate the report that contains all the information of license server, you can click the **Try it out** option, and then click **Execute**.

The license inventory report contains information such as details about the configuration of team space licensing, a list of *Team Space Owners* and their email addresses, information about licenses that are mapped to the license server, and the current connectivity status of the license server.

Depending on your privileges, you can perform the following tasks in Test Hub:

Privileges	Tasks	More information
<i>Team space Owner</i> OR <i>Server Administrator</i>	Read about licensing on Test Hub.	DevOps Test Hub licensing information on page 180
	Configure licenses.	Configuring licenses for team spaces on page 181
	Configure named users.	Adding users to the Named Users list on page 184
	Remove or replace named users.	Removing or replacing named users on page 185
Licensed user	Perform tasks or operations in a team space.	Tasks in a team space on page 173
	Viewing details about licenses.	Viewing license details of the team space on page 186

DevOps Test Hub licensing information

You can find information about the licensing on Test Hub.

To use the features offered by Test Hub, you must purchase the following licenses:

- Named user
- Concurrent Virtual Service

Named user licenses

As a *Team space Owner*, you can configure the License Server and add users to the **Named Users** list for the team spaces.

The **Named user** licenses are automatically self-acquired on a first-come-first-serve basis. The licenses are always available for the users who are added to the **Named Users** list. Also, those users can perform the project tasks in their team space even though the **Disable automatic self-acquisition of named user licenses** checkbox is cleared in the **License Configuration** page. However, the other users receive an error notification about the unavailability of the license when they try to perform the project tasks.



Remember: A single **Named user** license is used from the total number of **Named user** licenses when you perform most of the operations in Test Hub.



Important: You can perform only the following operations in team space without using a license:

- Configure the License Server
- Modify the user roles in a team space
- Delete the team space



Remember: The **Named user** license is returned to the license pool only under any of the following conditions:



- When the **Named user** is no longer a member of any team space.
- When the **Named user** is removed as a user of Test Hub.



Important: The **Named user** license can take a few minutes to return to the license pool.

Concurrent Virtual Service licenses

Test Hub requires an additional license to use the service virtualization feature. You must have sufficient **Concurrent Virtual Service** licenses to run instances of virtual services that are contained in API Suites on Test Hub.

A single **Concurrent Virtual Service** is used from the total number of **Concurrent Virtual Service** licenses when any of the following events occur:

- An instance of a virtual service is started by running a virtual service resource. A license is consumed for every running instance of the virtual service.
- An API suite is run that has a scenario that references a local virtual service and the scenario begins

The **Concurrent Virtual Service** license is returned to the license pool when the virtual service instance stops.

For example, consider you decided to have three team spaces including the initial team space, then you must configure the licenses required for each team space on the License Server based on your requirement. You must then use the URL and ID of those License Servers to configure the licenses for the team spaces.

Related information

[Configuring licenses for team spaces on page 181](#)

Configuring licenses for team spaces

As a *Team space Owner*, you can configure the License Server for the team spaces on the **License Configuration** page so that members of the team space can perform various operations such as creating a project, running tests, and so on.

Before you begin

You must have completed the following tasks:

- Configured the licenses that you purchased on the Licensing Server.
- Been assigned the role of a *Team space Owner* of the team space.


- Installed the product and signed up as a user on Test Hub.
- Optional: Created a team space. See [Creating a team space on page 171](#).

About this task

You must have a valid license to work with the team spaces that you create on Test Hub. You must provide a separate URL and ID of the License Server for the team spaces when you configure the License Server. If you provided the URL and ID of the License Server that is already configured for any other team space, then the configuration of the license for the team space fails with an error message. If you delete the team space, then the URL and ID of the License Server used for that team space is released. You can then use that for another team space.




Note: When you upgrade Test Hub from the previous version to the latest version, then the license configuration of the previous release applies to the initial team space. You can verify the information from the **License Configuration** page of the initial team space.

1. Log in to Test Hub.
2. Click the **Settings** icon  from the left navigation pane.
3. Open the team space for which you want to configure the license from the **Switch to** option.



Note: The **Switch to** option is displayed only if you have two or more team spaces.

4. Click the **Settings** icon , and then **License**.

Result

The **License Configuration** page is displayed.

5. Enter the values in the **License Server URL** and **License Server ID** fields to configure the licenses for the team space.

Result



You can view the following messages based on the value that you enter in the **License Server URL** and **License Server ID** fields:

Values	Messages
No value	Add the missing license details or enter the License Server URL.
Incorrect format of the License Server URL or License Server ID	This is not a valid format of the License Server URL or License Server ID.
Correct value	License Server is already in use by another team space owner

6. Perform the following actions to view or hide the details of the License Server:



Note: The default status of the **License Server URL** and **License Server ID** fields are hidden.

Fields	Icons	Action	Results
◦ License Server URL		Click	Test Hub shows the URL and ID of the License Server that you have configured.
◦ License Server ID		Click	Test Hub hides the URL and ID of the License Server that you have configured.

7. **Optional:** Click **Check connection** to verify the current status of License Server.

Based on the availability of License Server, you can view the following messages:

- License Server is available
- License Server is unreachable

8. Select the **Disable automatic self-acquisition of named user licenses** option if you do not want to self-acquire **Named users** license on a first-come-first-serve basis for your team space.

The following table describes the functionality of the **Disable automatic self-acquisition of named user licenses** option:

Option	When	Description
Disable automatic self-acquisition of named user licenses	Cleared <input type="checkbox"/>	Any member of a team space can get a license when there are sufficient licenses are available for the team space.
	Select- ed <input checked="" type="checkbox"/>	Only such users who are added to the Named Users list get a license to perform the tasks in their team space. If other users try to perform any project operations in any of the team spaces, they cannot get a license even if sufficient licenses are available.

9. Click **Configure** to apply the license configuration.

Results

You have configured the license for the team space.

The **License usage** section displays the number of **Named user** licenses that are currently in use. However, the **License usage** section does not display the number of **Concurrent Virtual Service** licenses that are currently in use.

The **License configuration** section displays the number of **Named user**, **Concurrent Virtual Service**, and **Studio** licenses that are configured on the License Server.

What to do next

You can perform any of the following tasks:

- Add users to the **Named Users** list. See [Adding users to the Named Users list on page 184](#).
- Add members to your team space. See [Adding members to a team space on page 998](#).



Adding users to the Named Users list

As a *Team space Owner* in HCL DevOps Test Hub (Test Hub), you can add users to the **Named Users** list so that the licenses are always available for those users.

Before you begin

You must have completed the following tasks:

- Been assigned the role of a *Team space Owner* of the team space.
- Optional: Created a team space. See [Creating a team space on page 171](#).
- Configured the license for the team space. See [Configuring licenses for team spaces on page 181](#)

1. Log in to Test Hub.
2. Click the **Settings** icon  from the left navigation pane.
3. Open the team space for which you want to add users from the **Switch to** option.
4. Click the **Settings** icon , and then **License**.

Result

The **License Configuration** page is displayed.

5. Enter the name or the email ID of the user in the **Add named users** field, and then select the name from the list that displays to add members to the **Named Users** list.



Note: You can press `Ctrl` to select multiple users simultaneously.

6. Click **Add** to make selected users as **Named users**.

Results

You have added users to the **Named Users** list.



Remember: The **License usage** section does not display the number of **Concurrent Virtual Service** licenses that are currently in use.



Removing or replacing named users

When you want to remove a user from the list of named users or you want to replace an existing user with another named user, you as a *Team space Owner* in HCL DevOps Test Hub (Test Hub), can perform these changes from the **License Configuration** page.

Before you begin

You must have completed the following tasks:



- Been assigned the role of a *Team space Owner* for the team space that you want to modify the users from the **Named Users** list.
- Added users to the list of named users for the team space on Test Hub.




1. Log in to Test Hub.
2. Click the **Settings** icon  from the left navigation pane.
3. Open the team space for which you want to remove or replace the users from the **Named users** list from the **Switch to** option.
4. Click the **Settings** icon , and then **License**.

Result

The **License Configuration** page is displayed.

5. Perform the actions described in the following table to remove or replace users from the **Named Users** list:

Tasks	Actions
Remove a user from the list of named users.	<p>Perform the following steps:</p> <ol style="list-style-type: none"> a. Locate the user that you want to remove from the Named Users list. b. Click the Menu icon  next to the name of the user, and then click Remove. c. Click Remove on the Remove named user dialog to complete the action. <p> Remember:</p> <ul style="list-style-type: none"> ◦ When the Disable automatic self-acquisition of named user licenses option is cleared, and if the user who is removed from the Named Users list tries to perform any project operation, then the name of the user is added back to the list. The name of the user is added only if sufficient licenses are available.

Tasks	Actions
	 <ul style="list-style-type: none"> ◦ When the Disable automatic self-acquisition of named user licenses option is selected and you remove the user from the Named Users list, then the following operations are impacted by the user who is removed from the list: <ul style="list-style-type: none"> ▪ The test runs scheduled by the user might fail to run. ▪ The test results which are in progress by the user might fail to record complete details. ▪ The user cannot perform any of the project operations that include the generation of offline tokens that are used for long-run tests. <p> Note: If the user is currently on Test Hub, then the license can be used while the current session remains active.</p>
Replace a user in the list of named users.	<p>Perform the following steps:</p> <ol style="list-style-type: none"> a. Locate the user from the Named Users list that you want to replace with another user. b. Click the Menu icon  next to the name of the user, and then click Replace. c. Enter the email ID or name of another user that you want to add to the Named Users list. d. Click Replace to complete the action.

Results

You have removed or replaced a user from the **Named Users** list.

Viewing license details of the team space

You can view license details of the team space to know information such as number of licenses that are configured, licensed users or details of the License Server on the **License Configuration** page.

Before you begin


The *Team space Owner* must have configured the licenses for the team spaces on HCL DevOps Test Hub (Test Hub).

About this task

You as a licensed user can view only the number of licenses configured, details of the licensed users on the **License Configuration** page. However, if you are a *Team space Owner* or *Server Administrator*, you can also view the URL and ID of the License Server that is configured for the team space.


1. Log in to Test Hub.
2. Perform one of the following actions described in the following table to view the license details of the team space:

Roles	Action
<i>Team space Owner</i> or <i>Server Administrator</i>	Perform Steps 3 on page 187 to 6 on page 187 .
Any other users	Perform Steps 3 on page 187 to 5 on page 187 .

3. Click the **Settings** icon  from the left navigation pane.
4. Open the team space for which you want to view the license details from the **Switch to** option.



Note: The **Switch to** option is displayed only if you have multiple team spaces.

5. Click the **Settings** icon , and then **License**.

Result



The **License Configuration** page is displayed with the following license information:

- The number of **Named user** licenses that are currently in use.
- The number of **Named user** licenses that are configured on the License Server.
- The number of **Concurrent Virtual Service** licenses that are configured on the License Server.
- The number of **Studio** licenses that are configured on the License Server.
- The status of License Server.
- Whether the self-acquisition of named user licenses feature is enabled or disabled.
- The name and email ID of the users that are configured as named users by the *Team space Owner*.



Remember: The **License usage** section does not display the number of **Concurrent Virtual Service** licenses that are currently in use.

6. Perform the following actions to view or hide the details of the License Server:

Fields	Icons	Action	Results
◦ License Server URL		Click	Test Hub shows the URL and ID of the License Server that you have configured.
◦ License Server ID		Click	Test Hub hides the URL and ID of the License Server that you have configured.

Results

You have viewed the details about the licenses for the team space.

Chapter 5. Test Author Guide

This guide describes how to create and edit test assets in HCL DevOps Test Hub (Test Hub), and then publish test assets to the remote Git repository that you have configured in your project. This guide is intended for testers and test managers.

Edit branch overview


When you want to create or modify a test resource in HCL DevOps Test Hub (Test Hub), you must use an Edit branch. An Edit branch is a temporary branch of the branch in the project repository into which you want to add the created resource or edited resource.

You must be a *Team Space Owner*, *Project Owner*, or *Member* with a *Tester* role to create or edit a test resource.

Before you can create or edit a test resource, you must select the branch in the project repository into which you want to add the created resource or the branch that contains the resource to edit.

After selection of the branch in the project repository, you must create an Edit branch, which is a temporary branch to contain the resources when you either create or edit a resource.

When you create the Edit branch, any member of the project can use your Edit branch while the branch is available. While you provide a name for the Edit branch, you must adhere to the rules of Git branch naming conventions. For more information, refer to the [Git](#) documentation. You can also view all the Edit branches from the **Branch** drop-down list when other members of the project create an Edit branch.

To differentiate the remote branch and Edit branch from the **Branch** drop-down list, an edit icon  and the initials of the member who created the Edit branch are shown for the Edit branch. After you create the Edit branch, you can also view the Edit branch by accessing the remote Git repository from a browser.

If you want to save the created or edited resource to the branch in the project repository, you must publish the changes from the Edit branch to the branch in the project repository. When the project repository is replicated automatically to the remote repository, the created or edited resources are then available in the remote repository in the branch they were published. The Edit branch that you used is automatically deleted after you publish the contents in the Edit branch. The Edit branch automatically disappears from the **Branch** drop-down list and from the remote Git repository when you perform any of the following actions:

- Publish changes to the remote Git repository
- Discard the changes made in all test assets



Note: After the Edit branch disappears, the **Branch** field is set to the previously selected branch name.

Data overview

When you want to create realistic test scenarios that can simulate real-world or practical situations, you can use the data fabrication feature in HCL DevOps Test Hub (Test Hub) to fabricate test data. You can also create or edit datasets that pass variable data to tests at test run time.

You can access the data fabrication feature and datasets from **Author > Data** in the Test Hub UI. The home page of **Data** provides easy access to the data definitions and datasets of projects in multiple branches. You can efficiently organize the Data definitions and datasets that you have created or modified in Test Hub.

Data definitions represent the data structures that are composed of one or more generators. Test Hub provides you the interface to create and manage Data definitions so that you can design data structures that closely resemble the real-world scenarios. You can configure generators, define rules, set restrictions and distributions if required, and generate test data using a Data definition. You can generate realistic and diverse test data for comprehensive testing scenarios.

Datasets provide tests with variable data during a test run. The test that uses a dataset at the run time replaces a value in the recorded test with variable test data that is stored in the dataset. In Test Hub, you can create a dataset and use its values during the run time to test various scenarios.

To conclude, you can use the data fabrication feature and datasets in Test Hub to provide tests with comprehensive and realistic test data.

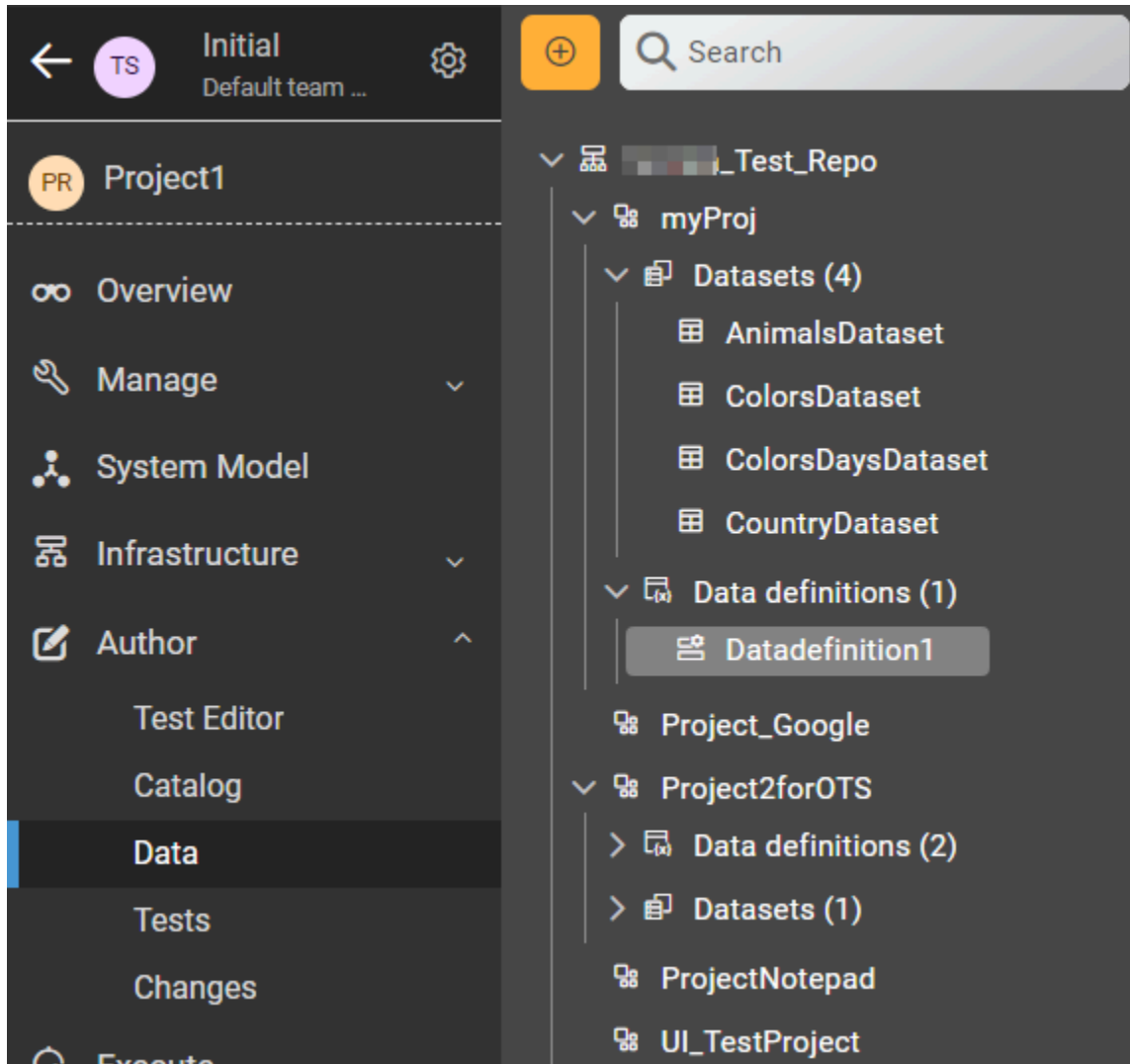
To use the dataset feature, see [Working with Datasets on DevOps Test Hub on page 216](#).

To use the data fabrication feature, see [Data fabrication on page 246](#).

Data navigator panel

The home page of **Data** contains a navigation panel called as the Data navigator panel. The Data navigator panel is a central hub for managing and organizing data definitions and datasets in multiple folders for your project.

Each folder in the Data navigator panel acts as a container for the existing Data definitions or datasets. Therefore, you can navigate and locate specific Data definitions or datasets within a folder of a project easily. Each folder can contain separate sections for the Data definitions or Datasets. You can click a Data definition or a dataset to open and manage the corresponding asset in the **Data** page for the project.



You can perform the following actions from **Author > Data**:

Actions	Descriptions
Search	You can search for a Data definition or a dataset in a branch by entering the name of the Data definition or dataset in the Search field.
View	You can view the contents of a Data definition or a dataset for a selected folder in a project. See Viewing a Data definition and its details on page 269 and Viewing a dataset on page 237 .
Create	You can create a Data definition or a dataset in a project. See Creating an empty Data definition on page 251 and Creating a dataset on page 217 .

Actions	Descriptions
Edit	You can modify a Data definition or a dataset when you want to run a test asset with different dataset values. See Modifying a Data definition on page 264 and Editing a dataset on page 226 .
Delete	You can delete a Data definition or a dataset when it is not required in your test environment. See Deleting Data definitions on page 273 and Deleting a dataset on page 240 .
Branch selection	<p>You can view the Data definitions and datasets of a project that are saved in multiple branches of the repository by selecting the name of the branch from the Branch list. When you access the Data definition or dataset page for the first time after adding the repository, the global branch is set to the default branch of the repository.</p> <p>When multiple repositories are added to the same project, the following events occur:</p> <ul style="list-style-type: none"> • The Data definitions and datasets that are stored in the selected branch of all the repositories in the project are displayed in the Data navigator panel. • The branches in each of the repositories are listed in the Branch list. • If a branch is not present in a repository, then it is not displayed in the Branch list.

**Notes:**

- Only a *Project Owner* or a project member whose role is a *Tester* can edit a Data definition or a dataset.
- To view the contents of an encrypted dataset, you must provide an encryption key that you set when you encrypted the dataset column.
- As a *Viewer*, when you try to modify a Data definition or dataset, an error message is displayed because you do not have the permission to modify them.

To conclude, the Data navigator panel provides easy access to projects, Data definitions, and datasets in multiple branches. You can efficiently organize the Data definitions and datasets in HCL DevOps Test Hub (Test Hub).

Prerequisites for working with Datasets and Data definition

You can find information about the tasks that you need to perform before you work with datasets and Data definitions in HCL DevOps Test Hub (Test Hub).

Management of classifications

You can create a classification and provide access only for the required members of a project in Test Hub. You can add the encrypted datasets, restricted database connections, or API secrets that you create in a project to the classification and ensure that it can be accessed and used only by the members who have access to the classification. See [Management of classifications on page 193](#).

Management of database connections

You can import the data that you have in an external database that can be used to create the test assets such as datasets and Data definitions in Test Hub.

You can establish and manage connections with the following external databases in **Test Hub**:

- Oracle
- PostgreSQL
- IBM DB2

See [Management of database connections on page 197](#).

Management of database queries

You can create a database query to import the data that is present in an external database, after you establish a connection with the external database. You can query the required data from a database when you create a database query to import the useful data present in your existing databases to create the test assets. See [Management of database query on page 206](#).

Management of classifications

You can find information about creating, editing, deleting, and granting access to a classification in HCL DevOps Test Hub (Test Hub).

You can find the following information about managing the classifications:

- [Creating a classification on page 193](#)
- [Editing a classification on page 194](#)
- [Deleting a classification on page 195](#)
- [Granting access to classification on page 196](#)

Creating a classification

As a project *Project Owner* or *Tester*, you must create a classification to organize encrypted datasets, external database connections, and API secrets so that project members can use them efficiently..

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Created a project in Test Hub. See [Managing DevOps Test Hub projects on page 1010](#).
- Configured the repository that contains the test assets in your project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

The **Security** page is displayed.

4. Click **Create Classification** and enter a name for the classification.

5. Click **Create**.

The classification is added to the **Classification** list.

What to do next

You can grant access to other project members to use the encrypted dataset. See [Granting access to classification on page 196](#).

Editing a classification

You might want to change the name of a classification that you created, you can do so by editing the classification.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Created a classification. See [Creating a classification on page 193](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

The **Security** page is displayed.

4. Click a classification that you want to edit from the list under **Classification**.

5. Click the **Edit** icon  in line with the classification.

The **Edit classification** dialog is displayed.

6. Edit the name of the classification, and then click **Save**.

Results

You edited the name for a classification in your project.

Deleting a classification

After you create a classification, you can delete a classification when it is not required in your test environment.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Created at least one dataset and encrypted the dataset with an encryption key. See [Dataset encryption on page 233](#).
- Created at least two classifications. See [Creating a classification on page 193](#).

About this task

You can delete a classification from your project any time after you create a classification. For example, you might want to delete the classification if the classification that was created earlier is not required.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

The **Security** page is displayed.

4. Click a classification that you want to delete from the list under **Classification**.



Note: Before deleting a classification associated with encrypted datasets, you must remove or move the encrypted datasets under it to another classification so that the delete icon is enabled.

5. Click the **Delete** icon .

The **Delete Classification** dialog is displayed.

6. Click **Delete** to confirm the deletion of the classification.

Results

You deleted a classification.

Granting access to classification

You can grant or revoke access to the classification in your project to individual members with different roles or all members with a specific role. Without access to the classification, members cannot view, create, edit, delete, or use the classification.

Before you begin

You must be a *Project Owner* and have completed the following tasks:

- Created a project in Test Hub. See [Managing DevOps Test Hub projects on page 1010](#).
- Added one or more users to your project. See [Adding users to a project on page 1017](#).
- Created at least one classification. See [Creating a classification on page 193](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

The **Security** page is displayed.

4. Click a classification in the list under **Classification** for which you want to grant the access.

5. Choose any of the following methods to grant access to a member:

Choose from:

- Select the checkbox for the specific role under **Grant Read Only Access to roles** to grant read-only access to the members of a specific role. For example, if you select **Tester** then all members in the project with a tester role are granted read-only access to the selected classification. You can select any role or all the roles listed.
- To grant read/write access to specific members, perform the following steps:
 - a. Click **Add Members** under **Members with Read/Write Access**.

The **Additional Members** dialog is displayed.

- b. Enter the name or email ID to search the user or navigate through the list and select the member you want to grant access. You can select multiple members in the list.



Important:




Irrespective of the role that the member (**Owner**, **Tester**, or **Viewer**) was assigned in the project, access to the classification has to be specifically granted to the members from the **Security** page.

c. Click **Add Members**.

You granted read/write access to the added members.

6. Choose any of the following methods to revoke access to a member:

Choose from:

- Clear the checkbox for the specific role under **Grant Read Only Access to roles**, to revoke read-only access to the members of a specific role. For example, if you clear the **Tester** checkbox then access for all members in the project with a tester role is revoked for the selected classification. You can select any role or all the roles listed.
- Click the **Delete** icon  in line with the member name that you want to remove the specific members from the **Members with Read/Write Access** list.



Notes:

- Any member with access to the classification can remove access to other members specifically added or of all members with a specific role.
- Members with access to the classification can remove themselves from the access list provided that there is at least one member in the list. After removing themselves, members cannot add themselves back to the access list and must be added by any of the other members in the list.

Results

You have granted or revoked the classification access to project members.

Management of database connections

You can add a database connection to a project and import the data from an external database in HCL DevOps Test Hub (Test Hub).

You can also restrict the database connection when you add it to a project and then add it to a classification to grant or restrict access for other members of the project. After you add a database connection you can create the database query that can be used to create test assets.

You can find the following information about adding the database connection and managing them in Test Hub:

- [Adding a database connection on page 198](#)
- [Adding a database connection to a classification on page 200](#)

- [Removing a database connection from a classification on page 202](#)
- [Editing a database connection on page 202](#)
- [Duplicating a database connection on page 204](#)
- [Deleting a database connection on page 206](#)

Adding a database connection

You can add a database connection to your project in HCL DevOps Test Hub (Test Hub) to import the data from an external database and create the test assets. You can restrict the project members from using this database connection by adding it to a Classification and providing permission only to the required members.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Ensured that you have a valid Java Database Connectivity (JDBC) URL and credentials for the database.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Infrastructure > Database Connections**.

Result

The **Database Connections** page is displayed.

4. Click **New Connection**.

Result

The **New Connection** dialog is displayed.

5. Perform the following actions in the **Details** tab:

- a. Enter a name for the database connection in the **Name** field.

- b. Select the database from the **Driver** list.

The following are the supported databases:

- Oracle
- PostgreSQL
- IBM DB2

- c. Enter the URL for the database in the **URL** field.

- d. Enter the username for the database in the **Username** field.

- e. Enter the password for the database in the **Password** field.
 - f. Enter the schema name for the database in the **Schema** field.
 - g. Add a description for the database connection in the **Description** field.
6. Select the **Restrict access using Classifications** checkbox if you want to restrict other members of the project from using this database connection.

Later you can grant access to the required members by adding the connection to a classification.



Note: The option, **Restrict access using Classifications** is not selected. This is the default selection.

7. Perform the following steps to add the SSL certificate for encrypting the data exchanged between Test Hub and the database.
- a. Click the **SSL** tab.
 - b. Click **Select File** and browse to the location on your computer where you have the Java Keystore File.
 - c. Select the file, and then click **Open**.

Result

The selected file name is displayed.

- d. Enter the password for the Java Keystore File in the **Password** field.
8. Click **Test** to test the connection with the database.

Result







A message is displayed that confirms the connection with database.

9. Click **Save** to save the database.

A message is displayed that confirms database connection is added.

You can view the following details of the database connection on the **Database Connections** page:

Column header	Description
Name	Displays the name of the database connection that you added.
URL	Displays the URL of the database connection that you added.
User	Displays the name of the user who added the database connection.

Column header	Description	
Description	Displays the description that you entered when you created the database connection.	
Actions	 or 	Click the lock icon to go to the Security page. You can grant access to the database connection by adding it to a classification.  Note: The unlock icon is displayed to indicate that the database connection is not restricted for other members of the project, and they can use database connection to create the test assets.
		Click the edit icon to edit the database connection. See Editing a database connection on page 202 .
		Click the duplicate icon to duplicate the database connection. See Duplicating a database connection .
		Click the delete icon to delete the database connection. See Deleting a database connection .

Results

You have added a database connection to your project.

What to do next

You can add the database to a classification to provide access to other members of the project, if you have restricted it when you created the database connection. See [Adding a database connection to a classification on page 200](#).

You can create a database query using the database connection that you added to your project. See [Creating a database query on page 207](#).

Adding a database connection to a classification

You can add the database connection to a classification to grant access to the classification for other project members in HCL DevOps Test Hub (Test Hub). After the access is provided, they can use the database connection to create the database query and import the data from the database.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Added at least one database connection and restricted the access to other members of the project. See [Adding a database connection on page 198](#)
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created at least one classification. See [Creating a classification on page 193](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.


The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

Result

The **Security** page is displayed.

4. Click a classification for which you want to add the database connection in the list under **Classification**.

5. Click the **Add Connection** icon  to select the database that you want to add to the classification.

Result


The **Add Connection** dialog is displayed.

6. Click the **Database Connection** field to view the available database connection in the project.

7. Identify and select the database connection that you want to add to the classification.

8. Enter the password that you use to connect to the database in the **Password** field.



Tip: You can click the view icon  to view the entered password and ensure that you entered the correct password.

9. Click **Add**.

Result

The database connection that is added to the classification is displayed.

Results

You have added a database connection to the classification.

What to do next

You can create a database query by using the database connection that you added to your project. See [Creating a database query on page 207](#).

Removing a database connection from a classification

You can remove a database connection from a classification when you are no longer required to provide access to other members of your project.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Added at least one restricted database connection to a classification. See [Adding a database connection to a classification on page 200](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

Result

The **Security** page is displayed.

4. Click a classification in the list under **Classification** from which you want to delete the database connection.
5. Click the **Delete** icon  in line with the database connection that you want to delete from the classification.

Result

The **Remove database connection** dialog is displayed.

6. Click **Remove**.



Note: Removing the database connection from the classification also removes the password stored in HCL DevOps Test Hub (Test Hub) for that database. You must enter the password again to gain access to the database connection while adding it to a different classification.

Results

You have removed the database connection from a classification.

Editing a database connection

You can edit the database connection, when you want to change the database details that you added earlier or if you want to update the SSL certificate file.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Added at least one database connection. See [Adding a database connection on page 198](#)
- Ensured that you have credentials for the database that you want to edit in a HCL DevOps Test Hub (Test Hub) project.

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Infrastructure > Database Connections**.

Result

The **Database Connections** page is displayed.

4. Click the Edit icon  inline with the database that you want to edit.

Result

The **Edit connection** dialog is displayed.

5. Perform the following actions in the **Details** tab:

a. Edit the details of the connection.

b. Enter the password for the database in the **Password** field.

c. Select the **Restrict access using Classifications** checkbox if you want to restrict other members of the project from using this database connection.

Later you can grant access to the required members by adding the connection to a classification.



Note: The option, **Restrict access using Classifications** is not select. This is the default selection.

6. Perform the following steps to change or add the SSL certificate and encrypt the data exchanged through the database.

a. Click the **SSL** tab.

b. Click the **x** icon to remove the existing Java Keystore file.

c. Click **Select File** and browse to the location on your computer where you have the Java Keystore File.

d. Select the file, and then click **Open**.

Result

The selected file name is displayed.

e. Enter the password for the Java Keystore File in the **Password** field.

7. Click **Test** to test the connection with the database.

Result

A message is displayed that confirms the connection with database.

8. Click **Save** to save the database connection.

A message is displayed that confirms database connection is modified and the edited database connection is displayed on the **Database Connections** page.

Results

You have edited the database connection in your project.

Duplicating a database connection

When you want create a similar database connection with different user credentials, to create the similar database connection and restrict or unrestricted it, or to create a similar database connection by encrypting the data exchanged between HCL DevOps Test Hub (Test Hub) and database, you can create a new database connection using an existing database connection.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Added at least one database connection. See [Adding a database connection on page 198](#).
- Ensured that you have a valid URL and credentials for the database that you want to add or duplicate in a Test Hub project.

About this task

You can duplicate database connection and create a new one in the following scenarios:

- Creating the same database connection with or without restricting the access to other members of the project.
- Adding the same database connection to a different classification which consists of different datasets, secrets and have access to different group of people.
- Adding the same database with different user credentials.

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Infrastructure > Database Connections**.

Result

The **Database Connections** page is displayed.

4. Click the **Duplicate** icon  inline with the database connection that you want to duplicate.

Result

The **Duplicate connection** dialog is displayed.

The value in the **URL** field, and the selections in the **Driver** field and the **Restrict access using Classifications** checkbox are auto-populated with the current values and selections in the database connection that you duplicate.

5. Perform the following actions in the **Details** tab:
 - a. Enter a different name for the database connection in the **Name** field.
 - b. Enter the username for the database in the **Username** field.
 - c. Enter the password for the database in the **Password** field.
 - d. Select the **Restrict access using Classifications** checkbox if you want to restrict other members of the project from using this database connection.

Later you can grant access to the required members by adding the connection to a classification.



Note: The option, **Restrict access using Classifications** is not select. This is the default selection.

6. Perform the following steps to change or add the SSL certificate for encrypting the data exchanged through the database.
 - a. Click the **SSL** tab.
 - b. Click the **x** icon to remove the existing Java Keystore file.
 - c. Click **Select File** and browse to the location on your computer where you have the Java Keystore File.
 - d. Select the file, and then click **Open**.

Result

The selected file name is displayed.

- e. Enter the password for the Java Keystore File in the **Password** field.
7. Click **Test** to test the connection with the database.

Result

A message is displayed that confirms the connection with database.

8. Click **Save** to save the database connection.

Result

A message is displayed that confirms database connection is added and the new database connection created by using the existing database connection is displayed on the **Database Connections** page.

Results

You have created a new database connection by using the existing database connection in your project.

Deleting a database connection

You can delete the database connection from a project when it is no longer required.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Added at least one database connection. See [Adding a database connection on page 198](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Infrastructure > Database Connections**.

Result

The **Database Connections** page is displayed.

4. Click the **Delete** icon  inline with the database connection that you want to delete.

Result

The **Delete Connection** dialog is displayed.

5. Click **Ok** to confirm the deletion.

Result

The **Connection deleted successfully** message is displayed.

Results

You have deleted a database connection from your project.

Management of database query

You can find information about creating, editing, duplicating, and deleting the database query in HCL DevOps Test Hub (Test Hub).

You can find the following information about creating and managing the database queries:

- [Creating a database query on page 207](#)
- [Editing a database query on page 209](#)
- [Duplicating a database query on page 211](#)
- [Deleting a database query on page 213](#)
- [Publishing changes in an Edit branch on page 214](#)

Creating a database query

When you have the data that is useful for creating the test assets is stored in a database, you can import the data from the database and create a database Query in HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Added at least one database connection. See [Adding a database connection on page 198](#)
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task

When you want to create any database query, you must have an Edit branch. You can create the Edit branch during the creation of the database query. All your dataset resides in the Edit branch until you publish or discard the changes.

After you create the database query, you can either publish your changes to a remote Git repository or you can discard the changes made to the database query. The Edit branch automatically disappears from the **Branch** list as well as from the remote Git repository and sets the global branch to the previously selected branch in the following conditions:

- When you publish changes to the remote Git repository.
- When you discard the changes made in the dataset.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.


The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Create a folder, if you want to use a new folder to contain the resource by performing the following steps:

a. Select the project or a logical folder in the Data navigator panel, right-click, and then click **Folder**.

Alternatively, select the project or a logical folder in the Data navigator panel, click the **Open action menu** icon , and then click **Folder**.

The **Create new folder** dialog is displayed.


b. Enter a name for the folder.

For example, enter `MyFolder` as the name of the folder.

c. Click **Create**.

The folder is created under the project and displayed in the Data navigator panel.

6. Perform any of the following actions:

- Click the **Open action menu** icon  in the Data navigator panel, and then click **Database Query**.
- Click **Add files** in the right panel, and then click **Database Query**.

The **Edit branch** dialog is displayed.

7. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.

8. Perform the following steps in the **Create database query** dialog:

a. Enter a name for the database query in the **Name** field.

For example, the name of the database query can be `my_db_query`.

b. Select a folder.

c. Click **Save**.

Result

A message is displayed that confirms database query is created.

The database query is displayed in the **Data** page. The database query is listed under the logical folder `Datasets` under the physical folder `MyFolder` that you created. The name of the database query is suffixed with the `.dtd.yaml` file extension.

For example, the file name of the database query can be `my_db_query.dtd.yaml`.

9. Perform the following steps in the **Data** page to select the table and query the data from the database:

- a. Select the database from the **Connection** list.
- b. Select the table in the selected database from the **Table** list.



Result

The data from the selected table is displayed.

- c. Select the required column of the table from which you want to import the data.
- d. Click **View** to switch between the **Visual** and **Code** view.



Note: Note: The default view is **Visual** view. In the **Code** view you can type the query to get data from the database.

- e. Click the **Preview** icon  to view the queried data from the database in tabular format in the bottom window.
- f. Click **Save** icon  to save the queried data from the database.

Results

You have created a database query in your project.

What to do next

Publish the database query to the Git repository so that other members of the project can use the database query. See [Publishing changes in an Edit branch on page 214](#).

You can create a dataset using the database query. See [Creating a dataset by using a database query on page 224](#)

You can create a Data definition using the database query. See [Creating a Data definition by importing fields from a database query on page 258](#).

Editing a database query

You can edit the database query when you want to query different data from the database or if you want to query the data from a different database.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created at least one database query. See [Creating a database query on page 207](#).

About this task

When you want to modify any database query, you must have an Edit branch. You can create the Edit branch during the creation of the database query. All your dataset resides in the Edit branch until you publish or discard the changes.

After you modify the database query, you can either publish your changes to a remote Git repository or you can discard the changes made to the database query. The Edit branch automatically disappears from the **Branch** list as well as from the remote Git repository and sets the global branch to the previously selected branch in the following conditions:

- When you publish changes to the remote Git repository.
- When you discard the changes made in the dataset.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Navigate to the logical folder **Datasets** in the Data navigator and select the database query that you want to edit.

The name of the database query is suffixed with the `.dtd.yaml` file extension.

Result

The database query is displayed on the **Data** page.

6. Click **Edit**.



Alternatively, navigate to the logical folder **Datasets** in the Data navigator panel, right click the database query that you want to edit, and then click **Edit**.

7. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
 - Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.
8. Perform the following steps on the **Data** page to select a different table or query the different data from the same database or different database:
- a. Select a different table, modify the query, or different connection.
 - b. Click **View** to switch between the **Visual** and **Code** view.



Note: The default view is **Visual** view. In the **Code** view you can type the query to get data from the database.

- c. Click the **Preview** icon  to view the queried data from the database in tabular format in the bottom window.
- d. Click the **Save** icon  to save the modifications on the database query

Results

You have edited a database query in your project.

What to do next

You can publish the database query to the Git repository so that other members of the project can use the database query. See [Publishing changes in an Edit branch on page 214](#).

Duplicating a database query

You can create a new database query by using the existing database query, when you want the same database query with a different name or only slight modification required in the existing database query.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created at least one database query. See [Creating a database query on page 207](#).

About this task

When you want to create a database query by using the existing database query, you must have an Edit branch. You can create the Edit branch during the creation of the database query. All your dataset resides in the Edit branch until you publish or discard the changes.

After you create a database query using the existing database query, you can either publish your changes to a remote Git repository or you can discard the changes made to the database query. The Edit branch automatically disappears from the **Branch** list as well as from the remote Git repository and sets the global branch to the previously selected branch in the following conditions:

- When you publish changes to the remote Git repository.
- When you discard the changes made in the dataset.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Navigate to the logical folder **Datasets** in the Data navigator, right click the database query that you want to save with a different name, and then click **Save As**.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.

7. Enter a different name for the database query in the **Name** field.


8. Select a folder in the **Location** field.

9. Click **Save**.

Result

A message is displayed that confirms database query is created.

The database query is displayed on the **Data** page. The new database query created is listed under the logical folder `Datasets` under the selected physical folder that you selected. The name of the database query is suffixed with the `.dtd.yaml` file extension.

10. Make any required changes to the new database query, and then click **Save** icon  to save the newly duplicated database query.

Results

You have created a database query using the existing database query in your project.

What to do next

You can publish the database query to the Git repository so that other members of the project can use the database query. See [Publishing changes in an Edit branch on page 214](#).

Deleting a database query

You can delete a database query when it is no longer required.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created at least one database query. See [Creating a database query on page 207](#).

About this task

When you want to delete any database query, you must have an Edit branch. You can create the Edit branch during the creation of the database query. All your dataset resides in the Edit branch until you publish or discard the changes.

After you delete the database query, you can either publish your changes to a remote Git repository or you can discard the changes made to the database query. The Edit branch automatically disappears from the **Branch** list as well as from the remote Git repository and sets the global branch to the previously selected branch in the following conditions:

- When you publish changes to the remote Git repository.
- When you discard the changes made in the dataset.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Navigate to the logical folder **Datasets** in the Data navigator and right click the database query that you want to delete, and then click **Delete**.

The name of the database query is suffixed with the `.dtd.yaml` file extension.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.

Result

The **Delete database query** dialog is displayed.

7. Click **Delete** to confirm the deletion.

Alternatively, you can select the checkbox and add comments in the **Description of change** field. You can then click **Delete and Publish** to delete the database query and publish all of the changes in the edit branch.

Result

A message is displayed that confirms database query is deleted.

Results

You have deleted a database query in your project.

What to do next

You can publish the database query to the Git repository so that other members of the project can use the database query. See [Publishing changes in an Edit branch on page 214](#).

Publishing changes in an Edit branch


After you create or modify a database query in a project in HCL DevOps Test Hub (Test Hub), you can publish all the changes that are in an Edit branch to the branch in the project repository. Then, other members of the project can view and use the test resources. The Edit branch automatically disappears from the **Branch** list of the **Data** page and the changes made to the project are merged into the previously selected branch name.


Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created or modified at least one database query.

About this task

When you create or modify multiple assets in a project, the **Changes** page lists all the changed assets in the selected branch. In this scenario, you cannot publish or discard a single change. You must either publish or discard all changes that are made to all the listed assets. The **Changes** page also provides you with information about the number of commits that the selected branch is ahead and behind. After you publish an asset to the Git repository, all the other project members who have access to that asset can view the updated asset. You can click the **Menu** icon  in each row of the assets to perform the following actions on the **Changes** page:

Options	Actions
Edit	Click Edit to make any updates to the asset before publishing.
Discard	Click Discard to remove the changes that you made to the asset.
Save as	<p>Click Save as to save a copy of the asset.</p> <p> Note:</p> <p>When you try to publish a modified asset with conflicting edits, an error about conflicting changes is displayed. However, you can use the Save as option to save a copy of the asset edits under a new asset name. You must discard the edits that you made to the original asset.</p> <p>For example, consider a scenario when Amy Smith and John Doe edit the same database query, and Amy Smith edits and publishes the database query. On the Changes page, a message that indicates the number of commits ahead and behind in the branch is displayed when John Doe tries to publish the same database query. You can then use the Save as option to resolve this issue.</p>

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Go to **Author > Changes**.
4. Select the required branch from the **Branch** list.

All the updated assets that are present in the selected branch are displayed on the **Changes** page.

5. **Optional:** Click the **(n) commits ahead** option, and then click **Reset** beside a commit to revert your changes for that asset.
6. **Optional:** Click the **(n) commits behind** option, and then click **Merge** to merge the changes made by other members of the project into the selected branch.
7. Click **Publish changes**.
The **Publish** dialog is displayed.
8. Enter a description for the changes that you performed to the asset, and then click **Publish**.
A message is displayed that notifies changes made to the Git repository.



Note:

The Edit branch automatically disappears from the **Branch** list as well as from the remote Git repository and sets the global branch to the previously selected branch. After the Edit branch disappears, the **Branch** field is set to the previously selected branch name.

Results

You have published the changes in an Edit branch to the branch in the project repository. The other members of the same project, who have access to the test resource can view and use the test resource.

Working with Datasets on DevOps Test Hub

You can find information about creating datasets in HCL DevOps Test Hub (Test Hub), applying a dataset classification, editing datasets, and managing datasets that are in your project repository.

A dataset contains variable data that are used by a test at test run time. You create the datasets and associate them with your tests in the desktop products. You can decide whether you want to use the dataset associated with the test to be used at run time or override that associated dataset with another dataset at run time. You can create datasets in the desktop products itself and select the datasets as overrides in the desktop products.

When you want to run those tests from Test Hub and you want the tests to use a different set of variable data, you can create datasets in Test Hub. You can then select them to override the dataset associated with the tests that are in the project repository.

You can create a dataset in Test Hub by using any of the following methods:

- Creating an empty dataset. See [Creating a dataset on page 217](#).
- Importing data from a CSV file. See [Creating a dataset by using a CSV file on page 219](#).
- Creating a dataset with fabricated data. See [Creating a dataset by using a Data definition on page 221](#).

When you want to edit the data in a dataset, you must open it for editing. See [Editing a dataset on page 226](#).

After you edit the dataset, you must publish the changes, so that the dataset is added to the project repository. See [Publishing changes in datasets on page 236](#).

If the data in your dataset contains sensitive information, you can encrypt the dataset columns. See [Encrypting a dataset column on page 233](#).

If a dataset has encrypted columns and you want to view an encrypted dataset, see [Viewing an encrypted dataset on page 241](#).

When you want to remove the encryption for dataset columns, you can decrypt the dataset column. See [Decrypting a dataset column on page 235](#).

You can assign a classification for encrypted datasets. See [Managing an encrypted dataset on page 243](#).

You can change the classification of an encrypted dataset. See [Changing classification for an encrypted dataset on page 242](#).

Creating a dataset

You can create datasets in HCL DevOps Test Hub (Test Hub) to replace the values of existing dataset with the values of new dataset during a test or schedule run.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task

The default dimension of the dataset that you create in Test Hub is 1 Row, 1 Column (1R X 1C) dataset. You can enter the number of columns and rows required for the dataset during the creation of dataset.

When you want to create or modify any dataset, you must have an Edit branch. You can create the Edit branch during the creation of dataset. All your dataset reside in the Edit branch until you publish or discard the changes.

After you create the dataset, you can either publish your changes to a remote Git repository or you can discard the changes made to the dataset. The Edit branch automatically disappears from the **Branch** list as well as from the remote Git repository and sets the global branch to the previously selected branch in the following conditions:

- When you publish changes to the remote Git repository.
- When you discard the changes made in the dataset.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.


3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Perform any of the following actions:

- Click the **Open action menu** icon  in the Data navigator panel, and then click **Dataset**.
- Click **Add files** in the right panel, and then click **Dataset**.


The **Edit branch** dialog is displayed.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.

7. Create a folder, if you want to use a new folder to contain the resource by performing the following steps:

- a. Select the project or a logical folder in the Data navigator panel, right-click, and then click **Folder**.

Alternatively, select the project or a logical folder in the Data navigator panel, click the **Open action menu** icon , and then click **Folder**.

The **Create new folder** dialog is displayed.

- b. Enter a name for the folder.

For example, enter `MyFolder` as the name of the folder.

- c. Click **Create**.


The folder is created under the project and displayed in the Data navigator panel.

8. Perform the following steps in the **Create Dataset** dialog:

- a. Enter a name for the Dataset.
- b. Select a folder.
- c. Click **Next**.

- d. Select the **Create an empty dataset** option from the **Select source** list.
- e. Enter the number of columns and rows required for the dataset that you want to create.
- f. Click **Create**.

The dataset is displayed on the **Data** page. The dataset created is listed under the logical folder `Datasets` under the physical folder `MyFolder` that you created.

9. Enter the variable data in the table to match the data that you want to use in tests.
10. Click the **Save** icon  to save the dataset.

Results

You created a dataset in your project.

What to do next

You can perform the following tasks:

- Add, modify, or remove data in the dataset. See [Editing a dataset on page 226](#).
- Publish the dataset to the Git repository so that other members of the project can use the dataset. See [Publishing changes in datasets on page 236](#).

Related information

[Creating a dataset by using a CSV file on page 219](#)

[Creating a dataset by using a Data definition on page 221](#)

Creating a dataset by using a CSV file

When you use variable data that is contained in a CSV file, you can import the data from the CSV file and create a dataset in HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Saved a CSV file that contains variable data on your computer.

About this task

When you want to create or modify any dataset, you must have an Edit branch. You can create the Edit branch during the creation of dataset. All your dataset reside in the Edit branch until you publish or discard the changes.

After you create the dataset, you can either publish your changes to a remote Git repository or you can discard the changes made to the dataset. The Edit branch automatically disappears from the **Branch** list as well as from the remote Git repository and sets the global branch to the previously selected branch in the following conditions:

- When you publish changes to the remote Git repository.
- When you discard the changes made in the dataset.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.


3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Perform any of the following actions:

- Click the **Open action menu** icon  in the Data navigator panel, and then click **Dataset**.
- Click **Add files** in the right panel, and then click **Dataset**.


The **Edit branch** dialog is displayed.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.

7. Create a folder, if you want to use a new folder to contain the resource by performing the following steps:

- a. Select the project or a logical folder in the Data navigator panel, right-click, and then click **Folder**.

Alternatively, select the project or a logical folder in the Data navigator panel, click the **Open action menu** icon , and then click **Folder**.

The **Create new folder** dialog is displayed.

- b. Enter a name for the folder.

For example, enter `MyFolder` as the name of the folder.

- c. Click **Create**.

The folder is created under the project and displayed in the Data navigator panel.

8. Perform the following steps in the **Create Dataset** dialog:

- a. Enter a name for the Dataset.
- b. Select a folder in the **Location** field.
- c. Click **Next**.
- d. Select the **Import from a CSV file** option from the **Select source** list.
- e. Click **Select File** and choose the CSV file that contains variable data to import into the dataset.
- f. Enable **First row contains headers** by using the toggle icon if the CSV file contains the header.
- g. Click **Create**.

The dataset is displayed on the **Data** page. The dataset created is listed under the logical folder `Datasets` under the physical folder `MyFolder` that you created.

Results

You have created the dataset by using the CSV file in your project.

What to do next

You can perform the following tasks:

- Add, modify, or remove data in the dataset. See [Editing a dataset on page 226](#).
- Publish the dataset to the Git repository so that other members of the project can use the dataset. See [Publishing changes in datasets on page 236](#).

Related information

[Creating a dataset on page 217](#)

[Creating a dataset by using a Data definition on page 221](#)

Creating a dataset by using a Data definition

When you created a Data definition in HCL DevOps Test Hub (Test Hub) to fabricate test data, you can create a dataset by using the Data definition.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created a Data definition. See [Creating an empty Data definition on page 251](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.


3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Perform any of the following actions:

- Click the **Open action menu** icon  in the Data navigator panel, and then click **Dataset**.
- Click **Add files** in the right panel, and then click **Dataset**.


The **Edit branch** dialog is displayed.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.

7. Create a folder, if you want to use a new folder to contain the resource by performing the following steps:

- a. Select the project or a logical folder in the Data navigator panel, right-click, and then click **Folder**.

Alternatively, select the project or a logical folder in the Data navigator panel, click the **Open action menu** icon , and then click **Folder**.

The **Create new folder** dialog is displayed.

- b. Enter a name for the folder.

For example, enter `MyFolder` as the name of the folder.

- c. Click **Create**.

The folder is created under the project and displayed in the Data navigator panel.

8. Perform the following steps in the **Create Dataset** dialog:

- a. Enter a name for the Dataset.
- b. Select a folder in the **Location** field.
- c. Click **Next**.
- d. Select the **Fabricate data** option from the **Select source** list.
- e. Select the Data definition to use from the **Select Data Definition** list.
- f. Enter the number of rows in the **Number of Rows** field.
- g. Enter a seed value in the **Seed** field.
- h. Click the **Fabricate headers** toggle to generate a header row.
- i. Click **Create**.

The dataset is displayed on the **Data** page. The dataset created is listed under the logical folder `Datasets` under the physical folder `MyFolder` that you created.

Results

You have created the dataset by using a Data definition in your project.

What to do next

You can perform the following tasks:

- Add, modify, or remove data in the dataset. See [Editing a dataset on page 226](#).
- Publish the dataset to the Git repository so that other members of the project can use the dataset. See [Publishing changes in datasets on page 236](#).

Related information

[Creating a dataset on page 217](#)

[Creating a dataset by using a CSV file on page 219](#)

Creating a dataset by using a database query

When you use variable data that is contained in a database, you can create a database query to import the required data from a database and use the data that is present in the database query to create a dataset in HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created at least one database query. See [Creating a database query on page 207](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.


3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Perform any of the following actions:

- Click the **Open action menu** icon  in the Data navigator panel, and then click **Dataset**.
- Click **Add files** in the right panel, and then click **Dataset**.


The **Edit branch** dialog is displayed.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.

7. Create a folder, if you want to use a new folder to contain the resource by performing the following steps:

- a. Select the project or a logical folder in the Data navigator panel, right-click, and then click **Folder**.

Alternatively, select the project or a logical folder in the Data navigator panel, click the **Open action menu** icon , and then click **Folder**.

The **Create new folder** dialog is displayed.

- b. Enter a name for the folder.

For example, enter `MyFolder` as the name of the folder.

- c. Click **Create**.

The folder is created under the project and displayed in the Data navigator panel.

8. Perform the following steps in the **Create Dataset** dialog:

- a. Enter a name for the Dataset.
- b. Select a folder in the **Location** field.
- c. Click **Next**.
- d. Select the **Use a Database Query** option from the **Select source** list.
- e. Select the database query to use from the **Select Database Query** list.
- f. Enter the number of rows in the **Number of Rows** field.
- g. Click **Create**.

Result

A message is displayed that confirms dataset is created.

The dataset is displayed on the **Data** page. The dataset created is listed under the logical folder `Datasets` under the physical folder `MyFolder` that you created.

Results

You have created the dataset by using a database query in your project.

What to do next

You can perform the following tasks:

- Add, modify, or remove data in the dataset. See [Editing a dataset on page 226](#).
- Publish the dataset to the Git repository so that other members of the project can use the dataset. See [Publishing changes in datasets on page 236](#).

Related information

[Creating a dataset on page 217](#)

[Creating a dataset by using a CSV file on page 219](#)

Editing a dataset

You can add, modify, remove, import, or export data from a dataset.

Before you begin

You must have created a dataset or configured a repository that contains the dataset.

About this task

If you are a project **Owner** or **Tester** in HCL DevOps Test Hub (Test Hub), you can perform basic tasks in the datasets by right-clicking any row, column, or cell in the dataset to organize your data in a better way. For example, you can perform tasks such as updating data in a cell, inserting or deleting rows and columns, or renaming column names.

After you edit the dataset, you can save the changes made to the dataset, and then you can publish the dataset to the Git repository. If you save and close the edited dataset, the **Changes** page lists the edited dataset and later you can publish it to the Git repository for other members to use.

When you have a CSV file that has data separated from a character, then you can import that CSV file into the dataset. You can select any of the following separator characters from the **Configure Dataset** window, and the selection can be the separator character that you used in the CSV file:

- Comma
- Semicolon
- Space
- Tab
- Other

Consider that you have the data in the CSV file in the following format:

Name;CCNum
John;1234 5678 1234 5678
Bob;1122 3344 5566 7788
Amy;2233 4455 6677 8899

When you import the CSV file in the dataset, and then select the separator value as **Semicolon**, the data in the dataset is displayed as follows:

	Name	CCNum
1	John	1234 5678 1234 5678
2	Bob	1122 3344 5566 7788
3	Amy	2233 4455 6677 8899

If you want the data in its original format, that is, a semicolon (;) character to separate the data, then you can choose any other separator value from the **Configure Dataset** dialog.



Note: The default separator value is **Comma**.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch that contains your project with the datasets that you want to edit.

The projects in the selected branch are displayed in the Data navigator panel. The datasets are listed under the folder you created to contain the dataset within a logical folder named `Datasets`.

5. Click the dataset in the data navigator.












The dataset opens in the right pane.









6. Click **Edit**.




7. Select your action to either select an existing Edit branch or create an Edit branch:


- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.



8. Perform the following actions to use the options available for editing:



Options	Actions
Find and Replace 	<p>To find:</p> <ol style="list-style-type: none"> Click the Find and Replace icon . Enter the content that you want to search in the Find field. Select any or all the following options to find the search content more effectively: <ul style="list-style-type: none"> Select the Case sensitive icon  to search the content that is the exact letter case of the content entered in the Find field. Select the Match entire cell contents icon  to search for cells that contain only the characters that you have entered in the Find field. Select the Search using regular expression icon  to search the pattern that matches strings. <p>For example, to search a cell that contains any number between 0 to 9, do the following:</p> <ol style="list-style-type: none"> Enter <code>\d</code> in the Find field. Select the Search using regular expression checkbox. Click Find. Click the Find icon . If the text is found, the cell containing that text is selected. Click the Find icon  again to find further instances of the search text. <p>To find and replace:</p> <ol style="list-style-type: none"> Click the Find and Replace icon . Enter the content that you want to search in the Find field. Enter the content that you want to replace in the Replace field. Select any or all the following options to find and replace the content more effectively: <ul style="list-style-type: none"> Select the Case sensitive icon  to find the content that is the exact letter case of the content entered in the Find field. Select the Match entire cell contents icon  to find and replace for cells that contain only the characters that you have entered in the Find and Replace fields. Select the Search using regular expression icon  to find and replace the pattern that matches strings.


Options	Actions
	<p>e. Click the Replace icon  to replace the individual instances.</p> <p>f. Click the Replace All icon  to replace every instance of the content throughout the dataset.</p>
Undo 	<p>a. Click the Undo icon .</p> <p>b. Select the recent changes from the list that you want to undo, and then click the list.</p> <p>The Undo option reverts the change you made in the dataset.</p>
Redo 	<p>a. Click the Redo icon .</p> <p>b. Select the recent changes from the list that you want to redo, and then click the list.</p>
Import 	<p>You can import variable data into a dataset from the following sources:</p> <ul style="list-style-type: none"> ◦ When there is a large amount of data stored in a CSV file. ◦ When you use the data fabrication feature to generate test data. <p>Importing data from a CSV file</p> <p>a. Click the Import icon .</p> <p>b. Select Import a CSV file from the drop-down list.</p> <p>c. Click Select file and choose the CSV file that contains variable data to import into the dataset.</p> <p>d. Enable the First row contains headers by using the toggle icon if your CSV file contains the header.</p> <p>e. Select one of the following options to append or overwrite data in the dataset:</p> <ul style="list-style-type: none"> ▪ Enable Overwrite existing data by using the toggle icon to add the rows and columns from the beginning of the dataset. ▪ Enable Append existing data by using the toggle icon to add rows and columns to the end of the dataset. <p>f. Click Import.</p> <p>Importing data by using data fabrication</p>

Options	Actions
	<p>a. Click the Import icon .</p> <p>b. Select Fabricate data from the drop-down list.</p> <p>c. Select a Data definition from the drop-down list.</p> <p>d. Enter the number of rows that you want to generate in the Number of Rows field.</p> <p>e. Enter a seed value in the Seed (optional) field.</p> <p>f. Enable Fabricate headers by using the toggle icon if your schema contains the headers.</p> <p>g. Select one of the following options to append or overwrite data in the dataset:</p> <ul style="list-style-type: none"> ▪ Enable Overwrite existing data by using the toggle icon to add the rows and columns from the beginning of the dataset. ▪ Enable Append existing data by using the toggle icon to add rows and columns to the end of the dataset. <p>h. Click Import.</p>
Export 	<p>You can export variable data from the dataset into a CSV file to reuse in future tests when required. You must have opened a dataset from which you want to export the data.</p> <p>Click the Export icon  to download the dataset as a CSV file.</p>
Set as current row	<p>During the test run, if you want variable data to be selected from a current row instead of the first row in a dataset, right-click any cell in a row and select Set as current row.</p> <p>When rows are deleted:</p> <p>If you delete any row between row 1 and the current row, the current row data is taken from the next row. For example, when you set the current row as 6, and then you delete any row between row 1 to row 6, the current row remains at row 6, but the content of row 7 is moved to row 6.</p> <p>When rows are inserted:</p> <p>If you insert any new row between row 1 to the current row, the current row data is taken from the previous row. For example, when you set the current row as 6,</p>


Options	Actions
	and then you insert any row between row 1 to row 6, the current row remains at row 6, but the content of row 5 is moved to row 6.
Dataset Configuration	<p>You can configure the dataset by clicking the Menu icon , and then selecting Configure. In the Configure Dataset dialog, you can change the row and column settings and configure the string values in the dataset that contains variable data for tests to use when they run.</p> <ul style="list-style-type: none"> ◦ Select any of the separator values that you used in the CSV file. <p>The available options are Comma, Semicolon, Space, Tab, and Other. In the CSV file, if you have any other separator characters other than the available options, then you can select the Other option, and then can specify a value.</p> <p>For example, if the data in the CSV file is separated by a character #, then select the Other option and enter # in the field.</p> <ul style="list-style-type: none"> ◦ Configure the following options to change the row and column settings: <ul style="list-style-type: none"> ▪ Column header - Use an up-down control button to increment or decrement the value of the column header. ▪ Data start point - Use an up-down control button to increment or decrement the value of the data starting pointer. ▪ Current row - Use an up-down control button to increment or decrement the value of the current row. ◦ Configure the following options to change the string values in the dataset: <ul style="list-style-type: none"> ▪ Treat as null - Enter a string value that is to be treated as null when running the test. ▪ Treat as empty - Enter a string value that is to be treated as empty when running the test.

Options	Actions
	<p>For example, when you run the test and the data 123 in the dataset is to be treated as empty, then you must specify 123 in the Treat as empty field.</p> <ul style="list-style-type: none"> ▪ Treat empty text as null - Select this field when you want the dataset that contains any blank cells, and the value of those blank cells to be interpreted as null. <p>◦ Click Update to apply the changes.</p>
Discard 	Click the Menu icon  and select Discard to discard the changes made to the dataset.

9. Click the **Save** icon  to save the changes made to the dataset or the **Save All** icon  to save all changes made to different datasets by using the same Edit branch.

 **Note:** if you are editing different files that include datasets or Data definitions simultaneously, you can select your action from the following options:

To...	Do this...
Keep your current file open and close the open files on the right.	Right-click, and then click the Close Tabs to right option.
Keep your current file open and close the open files on the left.	Right-click, and then click the Close Tabs to left option.
Keep your current file open and close all the other open files.	Right-click, and then click the Close Others option.
Close all open file tabs.	Right-click, and then click the Close All Tabs option.

10. Click the **Publish** icon  to publish the dataset to the Git repository.

Results

You have edited the dataset.

Dataset encryption

Encrypted datasets are useful when you want to run tests that contain confidential information such as a set of passwords or account numbers.

When you run a test in HCL DevOps Test Hub (Test Hub) that uses an encrypted dataset, then you must provide an encryption key for decrypting the encrypted data in columns so that the data can be used in the test. If the test uses data from the multiple encrypted dataset columns, you must enter the same encryption key for every encrypted dataset column that the test uses.

When you run the test that uses the dataset with the encrypted column, the value of the column is decrypted at a run time. The data in the column is sent as a clear-text string in requests to the server. The actual values of the encrypted dataset variables are not displayed in the test log. The test log displays asterisks for the encrypted dataset variables.

You can use only one encryption key to encrypt data in the columns in any dataset.

You can set the dataset classification (encrypted password) for a dataset. The dataset classification that you set is applied for the dataset if it is saved in different branches of the project repository and you need not set a password for the dataset in each branch. If you want to use different passwords, then you can create a different dataset by cloning the dataset that you want.



Important:

The encryption keys that you use to encrypt data in a dataset are not stored on the server nor can be retrieved from the server. Therefore, you must remember to store the encryption keys in a secure location. You must use the same encryption keys to perform the following operations:

- View the encrypted values
- Decrypt data
- Enable the use of the encrypted dataset during test runs

Encrypting a dataset column

To secure test data, you must encrypt datasets in HCL DevOps Test Hub (Test Hub). You can encrypt data in the columns of a dataset by using an encryption key. When you run a test that utilizes a dataset with encrypted variables, you must enter the encryption key for the encrypted column that the test uses.

Before you begin

You must have created a dataset. See [Creating a dataset on page 217](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch that contains your project with the datasets that you want to edit.

The projects in the selected branch are displayed in the Data navigator panel. The datasets are listed under the folder you created to contain the dataset within a logical folder named `Datasets`.

5. Click the dataset in the data navigator.

The dataset opens in the right pane.

6. Click **Edit**.

7. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.

8. Right-click any cell in a column that you want to encrypt and select **Encrypt column data**.

Result

The **Encrypt Column** window is displayed.

9. Enter an encryption key in the **Encryption Key** field to encrypt the data in the column.



Remember: When you have already encrypted other columns in the dataset, you must enter the same encryption key that you used previously. You can use only one encryption key to encrypt columns in a dataset.



Important:

The encryption keys that you use to encrypt data in a dataset are not stored on the server nor can be retrieved from the server. Therefore, you must remember to store the encryption keys in a secure location. You must use the same encryption keys to perform the following operations:




- View the encrypted values
- Decrypt data
- Enable the use of the encrypted dataset during test runs

10. Click **Encrypt Column**.

Result


Asterisks are displayed instead of actual data for the encrypted column.

11. Click the **Save** icon  to save the changes to the dataset.

Results

You have encrypted the dataset column in your project.

What to do next

You can publish the dataset to the Git repository so that other members of the project can use the dataset by clicking the **Publish** icon . See [Publishing changes in datasets on page 236](#).

Decrypting a dataset column

To view the content of an encrypted dataset in HCL DevOps Test Hub (Test Hub), you can decrypt the dataset column. Removing encryption from a dataset revokes the protection offered to the test data.

Before you begin

You must have created at least one dataset and encrypted the dataset with an encryption key. See [Creating a dataset on page 217](#) and [Encrypting a dataset column on page 233](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch that contains your project with the datasets that you want to edit.

The projects in the selected branch are displayed in the Data navigator panel. The datasets are listed under the folder you created to contain the dataset within a logical folder named `Datasets`.

5. Click the dataset in the data navigator.

The dataset opens in the right pane.

6. Click **Edit**.

7. Select your action to either select an existing Edit branch or create an Edit branch:
 - Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
 - Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.
8. Right-click encrypted cells that display the contents with asterisks, and then select **Decrypt column data**.


Result

The **Decrypt Column** window is displayed.

9. Enter the encryption key that you used to encrypt the data in the column in the **Encryption Key** field.
10. Click **Decrypt Column**.

Result


Asterisks are replaced with the actual data in the decrypted column.

11. Click the **Save** icon  to save the changes to the dataset.

Results

The encryption is removed from the selected column in the dataset. When you run a test that uses a dataset that contains decrypted data, the variable data is substituted for the original data in the recorded test without prompting for the encryption.

What to do next

You can publish the dataset to the Git repository so that other members of the project can use the dataset by clicking the **Publish** icon . See [Publishing changes in datasets on page 236](#).

Publishing changes in datasets

After you create, edit, or encrypt datasets in HCL DevOps Test Hub (Test Hub), you can publish all changes made to a dataset to a branch in the project. After you publish the dataset, other members in the project can use your dataset in their test run, if required. The Edit branch automatically disappears from the **Branch** list and changes made to the datasets are merged into the selected branch of the project repository.

Before you begin


You must have created, edited, or deleted dataset assets in Test Hub.

About this task

When you create or edit multiple datasets, the **Changes** page lists all datasets in the corresponding Edit branches. When you have many datasets listed on the **Changes** page, you can discard individual changes from the list by clicking **Discard** from **Actions** column, then publish or discard all changes made to datasets. If you do not modify any datasets, the **Changes** page displays a message that indicates there are no changes in any of the datasets.

The **Changes** page also provides you with information about the number of commits that the Edit branch is ahead and behind. You can click the **(n) commits ahead** option, and then click **Reset** to revert your changes. Similarly, you can also click the **(n) commits behind** option, and then click **Merge** to merge the changes of other member into the Edit branch.

You can publish the dataset to the remote Git repository either from the **Changes** page or from the dataset editor panel. When you modify the dataset and publish it to the Git repository, the other members who have access to that dataset can view the updated dataset.

You can click the **Open action menu** icon  to perform the following actions from the **Changes** page:

Options	Actions
Edit	Click Edit to make any updates to the dataset before publishing.
Save As	Click Save As to save a copy of the dataset.
Discard	Click Discard to remove the changes that you made to the dataset asset.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Go to **Author > Changes**.
4. Click **Publish changes**.
5. Enter a description for the changes made to the dataset, and then click **Publish**.

Result

A notification is displayed for successful pushing of changes to the Git repository.

Results

You have published the modified dataset into the Git repository.

Viewing a dataset

When test assets includes datasets, you can view the contents of a dataset from HCL DevOps Test Hub (Test Hub). The datasets residing in the Git repository are listed in the **Datasets** page.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Created or joined a project in your team space in Test Hub. See [Adding a project on page 1013](#).

- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created at least one dataset and encrypted the dataset with an encryption key. See [Creating a dataset on page 217](#) and [Encrypting a dataset column on page 233](#).
- Created a classification for an encrypted dataset. See [Managing an encrypted dataset on page 243](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch that contains your project with the datasets that you want to edit.

The projects in the selected branch are displayed in the Data navigator panel. The datasets are listed under the folder you created to contain the dataset within a logical folder named `Datasets`.

5. Click the dataset in the data navigator.

The dataset opens in the right pane.

Results

You have viewed the contents of the dataset.

Viewing an encrypted dataset

You can use the **Dataset** page to view the contents of an encrypted dataset from Test Hub.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created or joined a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created at least one dataset and encrypted the dataset with an encryption key. See [Creating a dataset on page 217](#) and [Encrypting a dataset column on page 233](#).
- Created a classification for an encrypted dataset. See [Managing an encrypted dataset on page 243](#).
- Granted classification access to project members. See [Managing an encrypted dataset on page 243](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch that contains your project with the datasets that you want to edit.

The projects in the selected branch are displayed in the Data navigator panel. The datasets are listed under the folder you created to contain the dataset within a logical folder named `Datasets`.

5. Select the encrypted dataset that you want to view.

The dataset is displayed in the Dataset panel with the data in the encrypted column masked with Asterisks (***)

6. Right-click the encrypted column and click **Show encrypted data**.

You can right-click the decrypted column and select **Hide encrypted data** to encrypt the data again.



Important: The **Show encrypted data** option is available only when you have added the encrypted dataset to the respective classification.

Results


You have viewed the contents of an encrypted dataset.

Changing classification for an encrypted dataset

To move the encrypted dataset from one classification to another dataset in Test Hub, you can change the classification for an encrypted dataset.

Before you begin

You must have created at least two or more classifications. See [Managing an encrypted dataset on page 243](#).

1. Go to the **Datasets** page and find an encrypted dataset for which you want to change the classification.
2. Right-click the dataset, and then click the **Security** icon .
3. In the **Change the classification for the Dataset** dialog:

- a. Choose the classification from the list.
- b. Enter the encryption key for the dataset that was set while encrypting the dataset column in the **Password** field.
- c. **Save** the classification details.



Remember: You can perform this task only if you know the encryption key set for the dataset while encrypting the dataset column.

Results

You have changed the classification for an encrypted dataset.

Related information

[Managing an encrypted dataset on page 243](#)

Deleting a dataset

You can delete the dataset when it is not required in your test environment.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Created at least one dataset in your Git repository that you have configured.

About this task

You can delete datasets in Test Hub in the following scenarios:

- Datasets that you created in the desktop clients and that are cloned to the repository in your project on Test Hub.
- Datasets that you created in Test Hub and that are published to the repository in your project.
- Datasets that are in the `.csv` file format.

You cannot delete the following datasets:

- Datasets that you created or edited in Test Hub and that are not published to the repository in your project.
- Datasets that are in the `.sit` file format.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch that contains your project with the datasets that you want to edit.

The projects in the selected branch are displayed in the Data navigator panel. The datasets are listed under the folder you created to contain the dataset within a logical folder named as `Datasets`.

5. Right-click the dataset in the data navigator, and then click the **Delete** option.
6. Select your action to either select an existing Edit branch or create an Edit branch:
 - Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
 - Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.
7. Perform the following steps in the **Delete <Dataset>** dialog that is displayed, where `<Dataset>` displays the name of the selected dataset.
 - a. Select the option to publish the deletion from the Edit branch to the main branch.
 - b. Click **Delete**.

The selected dataset is deleted and the deletion is published from the edit branch to the main branch.

Results


You deleted a dataset.


Viewing an encrypted dataset

You can use the **Dataset** page to view the contents of an encrypted dataset from HCL DevOps Test Hub (Test Hub).

Before you begin

You must have opened the dataset to view the contents.

1. Perform the following steps, if you have created a classification but yet to add the encrypted dataset to it:
 - a. Find the encrypted dataset that you are interested in, and then right-click it.
 - b. Click the **Security** option  in the menu.
The **Change the classification for the dataset** dialog is displayed.
 - c. Select a classification from the list and enter the encryption key set for the dataset.
 - d. Click **Save** to save the classification details.
2. Right-click the encrypted column and click **Show encrypted data**.
You can right-click the decrypted column and select **Hide encrypted data** to encrypt the data again.

 **Important:** The **Show encrypted data** option is available only when you have added the encrypted dataset to the respective classification.

Results


You have viewed the contents of an encrypted dataset.


Changing classification for an encrypted dataset

To move the encrypted dataset from one classification to another dataset in HCL DevOps Test Hub (Test Hub), you can change the classification for an encrypted dataset.

Before you begin

You must have created at least two or more classifications. See [Creating a classification on page 193](#).

1. Go to the **Datasets** page and find an encrypted dataset for which you want to change the classification.
2. Click the **Lock** icon  from the **Actions** column of a dataset.
3. In the **Change the classification for the Dataset** window:
 - a. Choose the classification from the list.
 - b. Enter the encryption key for the dataset that was set while encrypting the dataset column.
 - c. **Save** the classification details.

 **Remember:** You can perform this task only if you know the encryption key set for the dataset while encrypting the dataset column.

Results

You have changed the classification for an encrypted dataset.

Related information

[Managing an encrypted dataset on page 243](#)

[Creating a classification on page 193](#)

Managing an encrypted dataset

You can use encrypted datasets to limit access to confidential information such as account number or passwords. You can arrange data by an appropriate category so that project members can use datasets more effectively in certain tests and protect them.

About this task

A dataset can contain classified information that other members can access with permission. As a project owner, you can group encrypted datasets into different classifications and enable project members to view and edit datasets and run tests associated with the encrypted datasets. After you created a classification, you can change the classification for a dataset. You can also delete a classification if you do not require it in your test environment.



Notes:

- You must grant access and provide an encryption key of an encrypted dataset to other members of the project to work with the encrypted dataset.
- A project member who has been added as a *Tester* role can work with the encrypted dataset.

Adding an encrypted dataset to a classification

As a project *Project Owner* or *Tester*, you can organize encrypted datasets by adding it to a classification so that project members can use and protect datasets more efficiently.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Created a project in Test Hub. See [Managing DevOps Test Hub projects on page 1010](#).
- Configured the repository that contains the test assets in your project. See [Adding repositories to a server project on page 1014](#).
- Created at least one dataset and encrypted the dataset with an encryption key. See [Dataset encryption on page 233](#).
- Created at least one classification. See [Creating a classification on page 193](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

The **Security** page is displayed.


4. Click **Add Dataset** to select the encrypted dataset to become part of the classification.

The **Add Dataset** dialog is displayed.

5. Perform the following steps:

- a. Click the **Encrypted dataset** field to view the encrypted dataset.
- b. Identify and select the encrypted dataset that you wanted to add to the classification.
- c. Enter the encryption key that you used to encrypt the dataset in the **Password** field.



Tip: You can click the view icon  to view the entered encryption key and ensure that you entered the correct key.

- d. Click **Add**.

The encrypted dataset that is added to the classification is displayed.

- e. You can add other encrypted datasets to the same classification, if required.

What to do next

You can grant access to other project members to use the encrypted dataset. See [Granting access to classification on page 196](#).

Moving an encrypted dataset to another classification

When you add many encrypted datasets to the same classification, you can move some of them to another classification or when you want to delete a classification but you want to retain the encrypted datasets associated with it, you can move them to different classification before deleting the classification.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Created at least one dataset and encrypted the dataset with an encryption key. See [Dataset encryption on page 233](#)
- Created at least two classifications. See [Creating a classification on page 193](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

The **Security** page is displayed.

4. Click a classification from the list under **Classification** from which you want to move the encrypted dataset.

5. Click the **Edit** icon  inline with the encrypted dataset that you want to move to another classification.

The **Change the classification for the dataset** dialog is displayed.

6. Click a classification from the list under **Classification** to which you want to move the encrypted dataset.

7. Enter the encryption key that you used to encrypt the dataset in the **Password** field.

8. Click **Save**.

The encrypted dataset is moved to the classification that you selected and the same is displayed on the **Security** page.

Results

You have moved the encrypted dataset from one classification to another.

Removing a dataset from the classification

You can remove a dataset from a classification when it is no longer required.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Created at least one dataset and encrypted the dataset with an encryption key. See [Dataset encryption on page 233](#)
- Created a classification and added the encrypted dataset to it. See [Creating a classification on page 193](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

The **Security** page is displayed.

4. Click a classification from the list under **Classification** from which you want to delete the encrypted dataset.

5. Click the **Delete** icon  in line with the encrypted dataset that you want to delete from the classification.

The **Remove Dataset** dialog is displayed.

6. Click **Remove**.



Note: Removing dataset from the classification also removes the password stored in Test Hub for encrypted data. You must provide the password again while adding it to a different classification.

Results

You have removed the datasets from a classification.

Data fabrication

Test data is the core component of any application testing. It is very time-consuming to create test data manually. You can use the data fabrication feature in HCL DevOps Test Hub (Test Hub) to create realistic datasets for comprehensive testing scenarios. The feature provides data fabrication capabilities that can help you to generate synthetic data quickly and efficiently without the risk of data leaks or privacy issues.

Before you start using the data fabrication feature, you can review the information in [Overview of data fabrication on page 246](#).

You can access the data fabrication feature as **Data** in the Test Hub UI and you can generate the test data as the Comma-Separated Values (CSV) files that is saved in your system. You can then use these files to replace any existing datasets in a test during the run time.

The data fabrication feature in Test Hub provides the following advantages:

- Efficiency, which enables you to generate large volumes of realistic test data rapidly compared to manually creating datasets that can be error prone and time consuming.
- Data diversity, which enables you to define various data generation rules and distribution patterns so that you can fabricate diverse test data covering a wide range of scenarios.
- Data privacy, which enables you to fabricate information that is not based on real data and not breaching any issues of data privacy or sensitive data.
- Data re-usability, which enables you to reuse the fabricated data across multiple test cases and scenarios.

You can use the generated data during test runs. If a dataset is already associated with a test, you can replace the dataset at the test run time with the dataset that is created from the data fabrication feature of Test Hub. This allows you to integrate the generated test data with tests and ensures that your tests are run by using secure and realistic data. Thereby, you can enhance the effectiveness and efficiency of your testing efforts.

Overview of data fabrication

You can find information about an overview of the data fabrication feature that is available in HCL DevOps Test Hub (Test Hub).

Contents

- [Advantages offered by the data fabrication feature on page 247](#)
- [Data definitions on page 247](#)
- [Data definition catalog on page 247](#)
- [Using the data fabrication feature on page 248](#)

Advantages offered by the data fabrication feature

The data fabrication feature in Test Hub provides the following advantages:

- Efficiency, which enables you to generate large volumes of realistic test data rapidly compared to manually creating datasets that can be error prone and time consuming.
- Data diversity, which enables you to define various data generation rules and distribution patterns so that you can fabricate diverse test data covering a wide range of scenarios.
- Data privacy, which enables you to fabricate information that is not based on real data not breaching any issues of data privacy or sensitive data.
- Data re-usability, which enables you to reuse the fabricated data across multiple test cases and scenarios.

To use the data fabrication feature in Test Hub, you must be aware of the terms and concepts that are generally applied in data fabrication or test data generation.

Data definitions

Data definitions are fundamental building blocks that are used to define data structures and generate test data. You can use the data fabrication feature of Test Hub to create Data definitions and to specify the characteristics and attributes of the test data that you need for your testing scenarios. You can then use the Data definitions to generate synthetic test data that closely resemble real-world scenarios.

Data definition catalog

You can access the catalog for a project from **Author > Catalog** in the Test Hub UI. The **Catalog** page provides a structured overview of the available built-in generators and custom generators for a selected project. You can use the **Search catalog** field to quickly find the data generators by its name and also create new custom categories and new generators for a project in a branch.

The **Catalog** page displays a collection of built-in data generators in the **Basic** category. In addition to the **Basic** category of generators, the **Catalog** page displays all the custom categories and the generators under them, if any, for the project. You can use the basic and custom generators in Data definitions to generate common types of data. Because they are readily available for immediate use, you can quickly select and apply them in the data definitions of your project.

The **Catalog** tab contains the following types of generators:

- Basic generators
- Custom generators

Using the data fabrication feature

You can fabricate test data by using the data fabrication feature in Test Hub.

You can find the flow of the different tasks that you can perform to fabricate test data. See [Task flow: Generation of test data on page 249](#).

To begin with you must create a Data definition or an entity that contains the fields for each type of data.



Important: Test assets or resources cannot be created or edited directly in a branch of the project repository. A temporary branch of the branch in the repository called the Edit branch must be created as a container for assets or resources while you work with them. The contents or changes made by using the Edit branch must be committed or published to the branch in the project repository, which merges the changes to the branch in the repository and removes the Edit branch.

You must add at least one generator as a field in the Data definition to generate test data. You can select any generator from the available generators listed in the **Catalog** tab of a Data definition. The generators in turn contain different options and fields that generate data by using a built-in database in Test Hub. See [Basic generators on page 273](#).

After you add the generators to the Data definition, you can save the Data definition. See [Saving a Data definition by adding basic generators with their default settings on page 260](#).

You can generate test data by using the default settings of the basic generators that you added to a Data definition. See [Generation of test data on page 323](#).

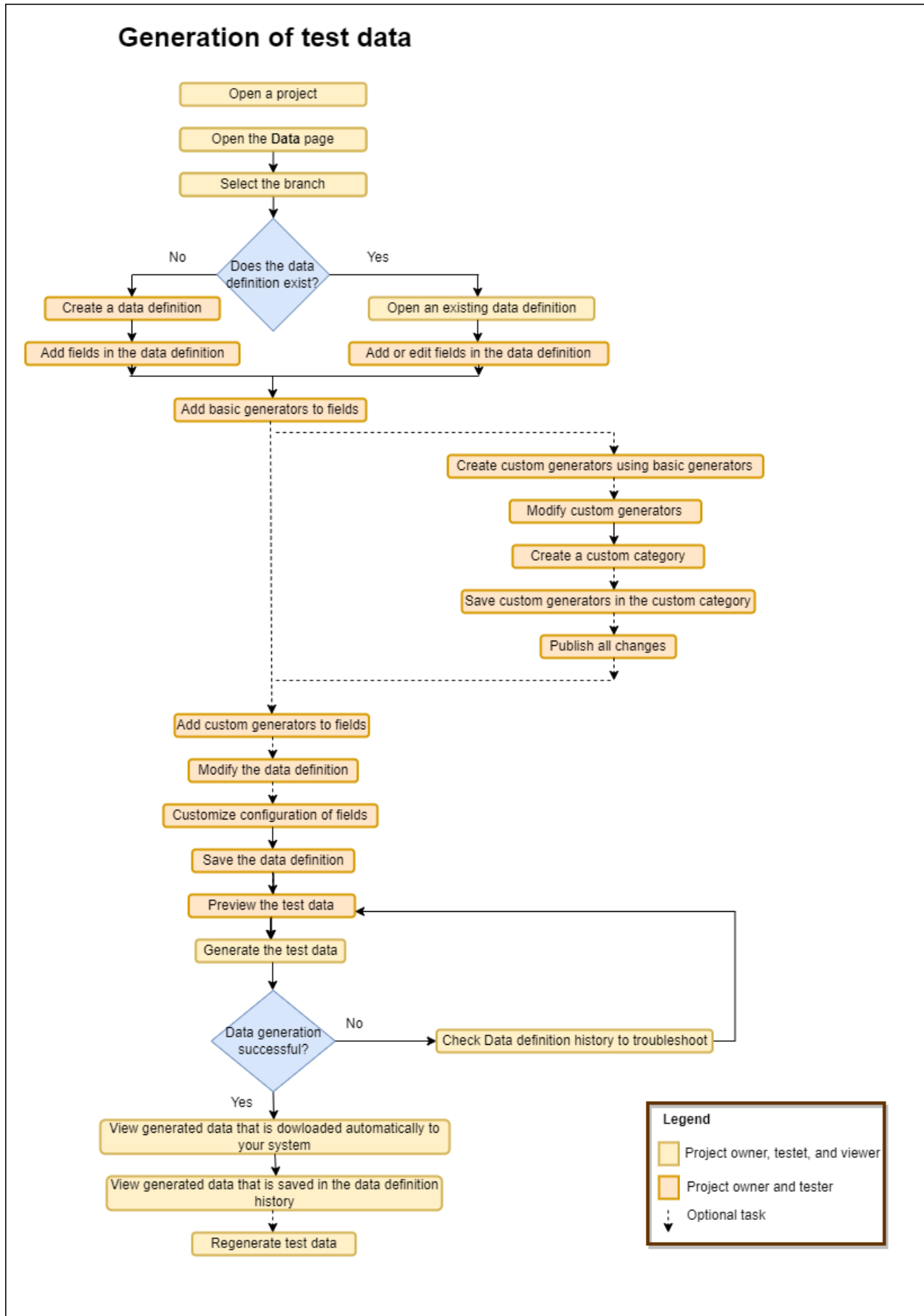
You can modify the default settings of the options and fields in a generator to suit your specific requirements of test data. You can only modify the settings and fields in a generator when it is added to a Data definition. The modified generators are applicable only to the Data definition in which they were modified. You cannot modify any of the built-in generators or save the modified basic generators in the **Basic** generators category in the **Catalog** tab. See [Saving a Data definition by adding basic generators with their modified settings on page 262](#).

Custom generators are modifications of the built-in generators that you modify any or all the fields in a generator to suit your specific requirements of your test data. The modified generator cannot be saved to the built-in generator and hence, you must save them as custom generators by providing a name to identify the custom generator. See [Management of custom generators on page 328](#).

You can create a custom category in the catalog to contain the custom generators that you create. See [Management of custom categories on page 335](#).

Task flow: Generation of test data

You can use the task flow diagram to get started with the generation of test data by using the data fabrication feature of HCL DevOps Test Hub (Test Hub). The following diagram can help you to visualize the sequence to easily generate the test data.



Creating an empty Data definition

You can create an empty Data definition in HCL DevOps Test Hub (Test Hub) as the first step to create a Data definition. After you save the empty Data definition to a folder in your project repository, you can edit the Data definition to add the different generators.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Read information about using Data definitions to fabricate test data. See [Overview of data fabrication on page 246](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.


3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Perform any of the following actions:

- Click the **Open action menu** icon  in the Data navigator, and then click **Data definition**.
- Click **Add files** in the right panel, and then click **Data definition**.


The **Edit branch** dialog is displayed.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:

- a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.
7. Create a folder, if you want to use a new folder to contain the resource by performing the following steps:

- a. Select the project or a logical folder in the Data navigator panel, right-click, and then click **Folder**.

Alternatively, select the project or a logical folder in the Data navigator panel, click the **Open action menu** icon , and then click **Folder**.

The **Create new folder** dialog is displayed.

- b. Enter a name for the folder.

For example, enter `MyFolder` as the name of the folder.

- c. Click **Create**.

The folder is created under the project and displayed in the Data navigator panel.

8. Perform the following actions in the **Create Data Definition** dialog:
- a. Enter a name for the Data definition.
 - b. Add a description about the Data definition in the **Description** field.
 - c. Select the folder in the **Location** field to store the Data definition.
 - d. Click **Next**.
 - e. Select the **Create an empty data definition** option from the **Select source** list.
 - f. Click **Create**.

The new Data definition opens in the **Data** page in the editing mode. The Data definition created is listed under the logical folder `Data definitions` under the physical folder `MyFolder` that you created.

Results

You created an empty Data definition in your project.

What to do next

You can perform the following tasks:

- Add basic generators with their default settings to the Data definition. You can then save and publish the Data definition. See [Saving a Data definition by adding basic generators with their default settings on page 260](#).
- Add basic generators to the Data definition, and then modify the configuration of the generators. You can then save and publish the Data definition. See [Saving a Data definition by adding basic generators with their modified settings on page 262](#).

Related information

[Creating a Data definition by importing fields from a CSV file on page 253](#)

[Creating a dataset by using a Data definition on page 221](#)

Creating a Data definition by importing fields from a CSV file

When you want to create a Data definition in HCL DevOps Test Hub (Test Hub) with specific fields for the test data, you can import a CSV file that defines the specific fields. After you import fields to the Data definition, you can edit the Data definition and add generators for the specific fields before you generate the test data.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Read information about using Data definitions to fabricate test data. See [Overview of data fabrication on page 246](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Ensured that a CSV file is available on your computer, from which you want to import data.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.


3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.


The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Perform any of the following actions:

- Click the **Open action menu** icon  in the Data navigator, and then click **Data definition**.
- Click **Add files** in the right panel, and then click **Data definition**.

The **Edit branch** dialog is displayed.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
 - Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.
7. Create a folder, if you want to use a new folder to contain the resource by performing the following steps:
- a. Select the project or a logical folder in the Data navigator panel, right-click, and then click **Folder**.
- Alternatively, select the project or a logical folder in the Data navigator panel, click the **Open action menu** icon , and then click **Folder**.
- The **Create new folder** dialog is displayed.
- b. Enter a name for the folder.
- For example, enter `MyFolder` as the name of the folder.
- c. Click **Create**.
- The folder is created under the project and displayed in the Data navigator panel.
8. Perform the following actions in the **Create Data Definition** dialog:
- a. Enter a name for the Data definition.
 - b. Add a description about the Data definition in the **Description** field.
 - c. Select the folder in the **Location** field to store the Data definition.
 - d. Click **Next**.
 - e. Select the **Import fields from a CSV file** option from the **Select source** list.
 - f. Click **Select file** and browse to the location on your computer where you have the CSV file.
 - g. Select the file, and then click **Open**.

The selected file name is displayed. The column headers identified in the CSV file are listed in the **File header** field.

h. Click **Create**.

The new Data definition opens in the **Data** page in the editing mode. The headers in the CSV file are added as the fields of the new Data definition but no generators are added. The Data definition created is listed under the logical folder `Data definitions` under the physical folder `MyFolder` that you created.

Results

You created a Data definition in your project, which has the fields from the CSV added as the fields of the new Data definition but no generators are added.

What to do next

You can perform the following tasks:

- Add basic generators with their default settings to the Data definition. You can then save and publish the Data definition. See [Saving a Data definition by adding basic generators with their default settings on page 260](#).
- Add basic generators to the Data definition, and then modify the configuration of the generators. You can then save and publish the Data definition. See [Saving a Data definition by adding basic generators with their modified settings on page 262](#).

Related information

[Creating an empty Data definition on page 251](#)

[Creating a dataset by using a Data definition on page 221](#)

Creating a Data definition by importing fields from a dataset

When you want to create a Data definition in HCL DevOps Test Hub (Test Hub) with specific fields for the test data, you can import the fields from a dataset. After you import fields to the Data definition, you can edit the Data definition and add generators for the specific fields before you generate the test data.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Read information about using Data definitions to fabricate test data. See [Overview of data fabrication on page 246](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).

- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Ensured that a dataset is available in your project repository, from which you want to import data.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.


3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.

5. Perform any of the following actions:

- Click the **Open action menu** icon  in the Data navigator, and then click **Data definition**.
- Click **Add files** in the right panel, and then click **Data definition**.


The **Edit branch** dialog is displayed.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
- Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.

7. Create a folder, if you want to use a new folder to contain the resource by performing the following steps:

- a. Select the project or a logical folder in the Data navigator panel, right-click, and then click **Folder**.

Alternatively, select the project or a logical folder in the Data navigator panel, click the **Open action menu** icon , and then click **Folder**.

The **Create new folder** dialog is displayed.

- b. Enter a name for the folder.

For example, enter `MyFolder` as the name of the folder.

- c. Click **Create**.

The folder is created under the project and displayed in the Data navigator panel.

8. Perform the following actions in the **Create Data Definition** dialog:

- a. Enter a name for the Data definition.
- b. Add a description about the Data definition in the **Description** field.
- c. Select the folder in the **Location** field to store the Data definition.
- d. Click **Next**.
- e. Select the **Import fields from a dataset** option from the **Select source** list.
- f. Search for the dataset in your project repository by entering the file name in the **Dataset** field.

Alternatively, browse to the location in your project repository, and then select the file.

- g. Click **Create**.

The new Data definition opens in the **Data** page in the editing mode. The headers in the dataset file are added as the fields of the new Data definition but no generators are added. The Data definition created is listed under the logical folder `Data definitions` under the physical folder `MyFolder` that you created.

Results

You created a Data definition in your project, which has the fields from the dataset added as the names of the generators but no generators are added.

What to do next

You can perform the following tasks:

- Add basic generators with their default settings to the Data definition. You can then save and publish the Data definition. See [Saving a Data definition by adding basic generators with their default settings on page 260](#).
- Add basic generators to the Data definition, and then modify the configuration of the generators. You can then save and publish the Data definition. See [Saving a Data definition by adding basic generators with their modified settings on page 262](#).

Related information

[Creating an empty Data definition on page 251](#)

[Creating a dataset by using a Data definition on page 221](#)

Creating a Data definition by importing fields from a database query

You can create Data definition to generate a random test data that is required to run a test in HCL DevOps Test Hub (Test Hub) using the specific database fields.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Read information about using Data definitions to fabricate test data. See [Overview of data fabrication on page 246](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created at least one database query. See [Creating a database query on page 207](#).

About this task

You must first create a database query and then create a Data definition using the database query to import the specific fields from the database query. After you import fields to the Data definition, you can edit the Data definition and add generators for the specific fields before you generate the test data.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.


3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch in the project repository from the **Branch** list.

The assets and resources in the selected branch are displayed in the Data navigator panel.


5. Perform any of the following actions:

- Click the **Open action menu** icon  in the Data navigator, and then click **Data definition**.
- Click **Add files** in the right panel, and then click **Data definition**.

The **Edit branch** dialog is displayed.

6. Select your action to either select an existing Edit branch or create an Edit branch:

- Perform the following steps to select an existing Edit branch:
 - a. Select an Edit branch from the **Select branch** list if Edit branches already exist in the selected branch of the repository.
 - b. Click **Ok**.
 - Perform the following steps to create the Edit branch:
 - a. Select **Create new branch**.
 - b. Enter a name for the Edit branch.
 - c. Click **Save**.
7. Create a folder, if you want to use a new folder to contain the resource by performing the following steps:
- a. Select the project or a logical folder in the Data navigator panel, right-click, and then click **Folder**.

Alternatively, select the project or a logical folder in the Data navigator panel, click the **Open action menu** icon , and then click **Folder**.

The **Create new folder** dialog is displayed.
 - b. Enter a name for the folder.

For example, enter `MyFolder` as the name of the folder.
 - c. Click **Create**.

The folder is created under the project and displayed in the Data navigator panel.
8. Perform the following actions in the **Create Data Definition** dialog:
- a. Enter a name for the Data definition.
 - b. Add a description of the Data definition in the **Description** field.
 - c. Select the folder in the **Location** field to store the Data definition.
 - d. Click **Next**.
 - e. Select the **Import fields from a database query** option from the **Select source** list.
 - f. Select the required database query file in your project, and then click **Create**.
- Result**
A message is displayed that confirms Data definition is created.

The new Data definition opens in the **Data** page in the editing mode. The columns in the database query are added as the fields of the new Data definition but no generators are added. The Data definition created is listed under the logical folder `Data definitions` under the physical folder `MyFolder` that you created.

Results

You have created a Data definition in your project by importing fields from a database query.

What to do next

You can perform the following tasks:

- Add basic generators with their default settings to the Data definition. You can then save and publish the Data definition. See [Saving a Data definition by adding basic generators with their default settings on page 260](#).
- Add basic generators to the Data definition, and then modify the configuration of the generators. You can then save and publish the Data definition. See [Saving a Data definition by adding basic generators with their modified settings on page 262](#).

Related information

[Creating an empty Data definition on page 251](#)

[Creating a dataset by using a Data definition on page 221](#)

Saving a Data definition by adding basic generators with their default settings

After you create a Data definition in HCL DevOps Test Hub (Test Hub), you must add the basic generators with their default settings and save the Data definition before you can generate test data.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Read information about using Data definitions to fabricate test data. See [Overview of data fabrication on page 246](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created an empty Data definition that is opened for editing.

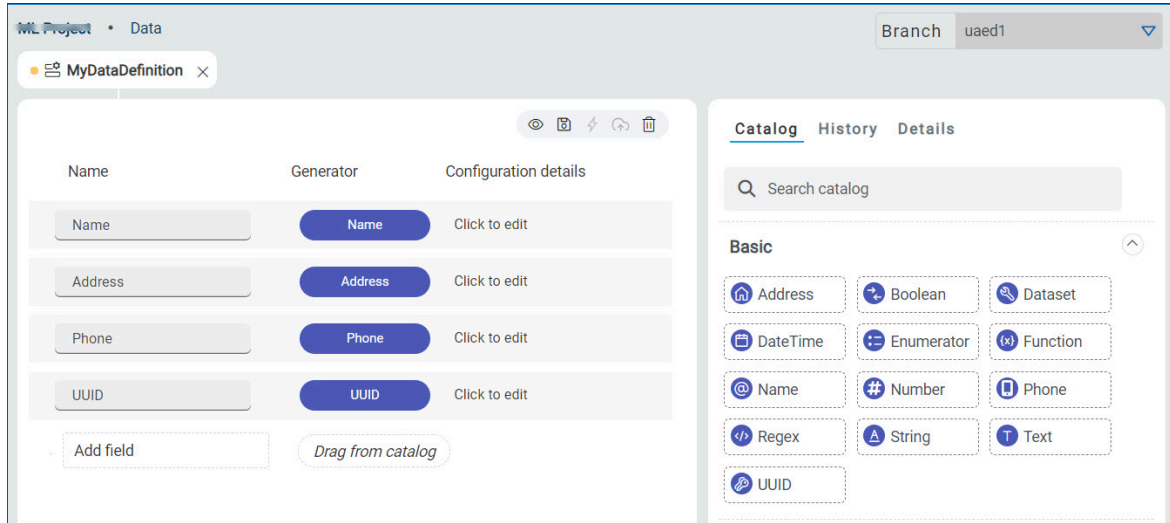
1. Drag a basic generator from the **Catalog** panel to the Data definition.



The basic generator is added as a field in the Data definition.

2. Drag other basic generators from the **Catalog** panel to the Data definition to complete the data structure that you want in the fabricated data.

Do not change any of the fields in the Data definition or the configuration of the basic generators.

For example, if you want to fabricate data about persons with their name, address, phone number, and a unique ID for each name or record, you can drag the **Name**, **Address**, **Phone**, and **UUID** generators to the Data definition, without making any changes to any of the fields in the Data definition or in the basic generators.





3. Click the **Save** icon  to save the changes made to the Data definition or the **Save All** icon  to save all changes made to different Data definitions by using the same Edit branch.




Note: When you edit different files that include datasets or Data definitions simultaneously, you can select your action from the following options:

To...	Do this...
Keep your current file open and close the open files on the right.	Right-click, and then click the Close Tabs to right option.
Keep your current file open and close the open files on the left.	Right-click, and then click the Close Tabs to left option.
Keep your current file open and close all the other open files.	Right-click, and then click the Close Others option.

	To...	Do this...
	Close all open file tabs.	Right-click, and then click the Close All Tabs option.

 **Note:** The changes you made are only visible and available to you in the Edit branch. You must publish the changes in the Edit branch to the branch in the project repository for the changes to be visible to the other members of the project.



4. Click the **Publish** icon  to publish the changes made to the Data definition to the branch in the project repository.

Results

You edited and saved a Data definition after adding basic generators with their default settings.

What to do next

You can perform any of the following tasks:

- Preview the test data generated by using the generators added as fields in the Data definition by clicking the **Preview** icon .
- Generate the test data by using the generators added as fields in the Data definition by clicking the **Generate** icon .

Saving a Data definition by adding basic generators with their modified settings

After you create an empty Data definition in HCL DevOps Test Hub (Test Hub), you must add the basic generators with their modified settings, and save the Data definition before you can generate test data.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Read information about using Data definitions to fabricate test data. See [Overview of data fabrication on page 246](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created an empty Data definition that is opened for editing.

1. Drag a basic generator from the **Catalog** panel to the Data definition.

The basic generator is added as a field in the Data definition.



2. Drag other basic generators from the **Catalog** panel to the Data definition to complete the data structure that you want in the fabricated data.
3. Perform the following steps to modify the parameter configurations of a basic generator in a field of the Data definition:

- a. Click a generator in the **Generator** column of a field where you want to customize the configuration. The parameters of the generator are displayed in the **Generator** dialog.

- b. Change the parameter settings in the **Generator** dialog to customize the generator.

- c. Close the **Generator** dialog.


The changes in the generator configuration are automatically saved.


4. Click the **Save** icon  to save the changes made to the Data definition or the **Save All** icon  to save all changes made to different Data definitions by using the same Edit branch.




Note: When you edit different files that include datasets or Data definitions simultaneously, you can select your action from the following options:

To...	Do this...
Keep your current file open and close the open files on the right.	Right-click, and then click the Close Tabs to right option.
Keep your current file open and close the open files on the left.	Right-click, and then click the Close Tabs to left option.
Keep your current file open and close all the other open files.	Right-click, and then click the Close Others option.

	To...	Do this...
	Close all open file tabs.	Right-click, and then click the Close All Tabs option.

 **Note:** The changes you made are only visible and available to you in the Edit branch. You must publish the changes in the Edit branch to the branch in the project repository for the changes to be visible to the other members of the project.



5. Click the **Publish** icon  to publish the changes made to the Data definition to the branch in the project repository.

Results

You edited and saved a Data definition after adding basic generators and then modifying the configuration of the generators.

What to do next

You can perform any of the following tasks:

- Preview the test data generated by using the generators added as fields in the Data definition by clicking the **Preview** icon .
- Generate the test data by using the generators added as fields in the Data definition by clicking the **Generate** icon .

Modifying a Data definition

After you create and publish a Data definition in HCL DevOps Test Hub (Test Hub), you can modify the fields or the configuration of the generators in an existing Data definition when you want to change the structure of the test data.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a Data definition. See [Creating an empty Data definition on page 251](#).

About this task

You can modify a Data definition after you open it for editing. After you make the modifications, you can save the Data definition, and either preview or generate test data with the changed data structure. If you want to commit the changes you made to the Data definition, you must publish the changes to the project repository. When you publish

the changes they become visible to other project members. This way, other project members can access and use from the updated Data definition. When you or other members of the project edit the same Data definition, you can view an icon with the initials of the member in the Data definition page.

The **Changes** page also lists the conflicting edits when another member publishes their edited Data definition first. When a member with conflicting edits tries to publish the edited Data definition, an error about conflicting changes is displayed. However, the member can use the **Save as** option in the **Changes** page to save and publish a copy of the Data definition with a different name. The member must then discard the edits that were made to the original Data definition.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.



3. Click **Author > Data**.




The **Data** page and the Data navigator panel are displayed.



4. Click the Data definition that you want to edit in the Data navigator panel.


The selected Data definition opens in the **Data** page.

5. Perform the actions that are listed in the following table to modify the Data definition:


Modification	Action
Addition of new fields	<p>Click the Add field option, and then enter a name for the field in the Name column.</p> <p> Note: A new row with an empty field is automatically displayed below the row that you edit.</p>
Addition of generators in fields	<p>Drag a generator from the Catalog panel into a field under the Generator column.</p> <p> Note: You can use the Search catalog field in the Catalog panel to find the required generator.</p>
Replacement of generators in fields	<p>Drag a generator from the Catalog panel into an existing generator field under the Generator column. The existing generator in the field is automatically replaced by the generator.</p>


Modification	Action
Modification of generator configurations	Click a generator in the Generator column. The generator configurations are displayed. Modify the options and settings in the generator configuration.  Note: If you modify the configurations of a basic generator, the changes are applicable only when the generator is used as field in a Data definition. The basic generator that is available in the catalog is not modified nor saved.
Inclusion of additional configuration details	Click the Add new configuration tab. Provide a name, and enter or select the other settings as an additional configuration to the generator.
Arrangement of fields	Drag and drop the field rows in the Data definition panel to rearrange the fields in the order you want.
Deletion of fields	Click the delete icon  in the row of a field to remove the field from the Data definition.
Deletion of Data definition	Click the Delete icon  in the Data definition panel to delete the Data definition from the existing location.


6. Click the **Save** icon  to save the changes made to the Data definition or the **Save All** icon  to save all changes made to different Data definitions by using the same Edit branch.

 **Note:** When you edit different files that include datasets or Data definitions simultaneously, you can select your action from the following options:

To...	Do this...
Keep your current file open and close the open files on the right.	Right-click, and then click the Close Tabs to right option.
Keep your current file open and close the open files on the left.	Right-click, and then click the Close Tabs to left option.
Keep your current file open and close all the other open files.	Right-click, and then click the Close Others option.

	To...	Do this...
	Close all open file tabs.	Right-click, and then click the Close All Tabs option.

 **Note:** The changes you made are only visible and available to you in the Edit branch. You must publish the changes in the Edit branch to the branch in the project repository for the changes to be visible to the other members of the project.

7. Click the **Publish** icon  to publish the changes made to the Data definition to the branch in the project repository.

Results

You modified an existing Data definition in your project.

What to do next

You can generate test data by using the modified Data definition. See [Generation of test data on page 323](#).

Publishing changes in an Edit branch


After you create or modify a Data definition, a generator, or a generator category in a project in HCL DevOps Test Hub (Test Hub), you can publish all the changes that are in an Edit branch to the branch in the project repository. After which, other members in the project can view and use the test resources. The Edit branch automatically disappears from the **Branch** list of the **Data** page and the changes made to the project are merged into the previously selected branch name.


Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a Data definition. See [Creating an empty Data definition on page 251](#).

About this task

When you create or modify multiple assets in a project, the **Changes** page lists all the changed assets in the selected branch. In this scenario, you cannot publish or discard a single change. You must either publish or discard all changes that are made to all the listed assets. The **Changes** page also provides you with information about the number of commits that the selected branch is ahead and behind. After you publish an asset to the Git repository, all the other project members who have access to that asset can view the updated asset. You can click the **Menu** icon  in each row of the assets to perform the following actions in the **Changes** page:

Options	Actions
Edit	Click Edit to make any updates to the asset before publishing.
Discard	Click Discard to remove the changes that you made to the asset.
Save as	<p>Click Save as to save a copy of the asset.</p> <p> Note:</p> <p>When you try to publish a modified asset with conflicting edits, an error about conflicting changes is displayed. However, you can use the Save as option to save a copy of the asset edits under a new asset name. You must discard the edits that you made to the original asset.</p> <p>For example, consider a scenario when Amy Smith and John Doe edit the same Data definition, and Amy Smith edits and publishes the Data definition. On the Changes page, a message that indicates the number of commits ahead and behind in the branch is displayed when John Doe tries to publish the same Data definition. You can then use the Save as option to resolve this issue.</p>

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Go to **Author > Changes**.

4. Select the required branch from the **Branch** list.

All the updated assets that are present in the selected branch are displayed in the **Changes** page.

5. **Optional:** Click the **(n) commits ahead** option, and then click **Reset** beside a commit to revert your changes for that asset.
6. **Optional:** Click the **(n) commits behind** option, and then click **Merge** to merge the changes made by other members of the project into the selected branch.
7. Click **Publish changes**.
The **Publish** dialog is displayed.
8. Enter a description for the changes that you performed to the asset, and then click **Publish**.
A notification is displayed for successful pushing of changes to the Git repository.

**Note:**

The Edit branch automatically disappears from the **Branch** list as well as from the remote Git repository and sets the global branch to the previously selected branch. After the Edit branch disappears, the **Branch** field is set to the previously selected branch name.

Results

You published the changes in an Edit branch to the branch in the project repository. The other members of the same project, who have access to the test resource can view and use the test resource.

Viewing a Data definition and its details

You can view the contents of a Data definition on the Data page in HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a Data definition. See [Creating an empty Data definition on page 251](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch from the **Branch** list.










The navigation panel displays all the existing projects in the selected branch.






5. Find the Data definition that you want to view, and then click it.

You can either enter the name of the Data definition in the **Search** field or expand the project and manually search in the Data navigator panel.

The selected Data definition opens in the **Data** page and the following elements are displayed:





Element	Description
The Name column	A list of fields and their custom names that exist in the Data definition is displayed in the column.

Element	Description
The Generator column	<p>The generator that is selected for the corresponding field is displayed in the column.</p> <p> Note: If any of the fields does not have a corresponding generator, then you cannot preview or generate the test data for the Data definition.</p>
The Configuration details column	<p>The configuration details of each field are displayed in the column.</p> <p> Note: The column displays details only when additional configuration is added for the field. Also, if you have added three or more additional configurations, then the ellipses icon  is displayed. You can hover over the icon to view the list of additional configuration names along with their distribution values.</p> <p>You can also click the column of a field to view the existing configuration details for the field in the corresponding <i>Generator</i> dialog.</p>
Initials of members	<p>The colored avatars, that represent the initials of the project members who are currently editing the data definition, are displayed in the page. You can hover over an avatar to view the name of the member and the branch that they are currently editing. This information can facilitate collaboration and communication among team members who are working on the same Data definition simultaneously. To understand how you can resolve conflicting edits, see Modifying a Data definition on page 264.</p>
The Preview icon 	<p>You can click the icon to preview the test data in the preview dialog of the Data page. The preview dialog contains the generate icon  and the close icon .</p> <p> Notes:</p> <ul style="list-style-type: none"> ◦ If you are in an Edit branch, then you can access the Generate icon  in the preview dialog only if the Data definition is saved. ◦ If you modify the Data definition but not save it, you can still preview the corresponding test data. However, you can generate the test data only after you save the Data definition.
The Save icon 	<p>You can click the icon to save the Data definition. If you modified the Data definition and you want to continue using the modified Data definition to generate test data without publishing it, you must save the changes. However, if you want</p>

Element	Description
	to collaborate with other members of your project, you must publish the Data definition.
The Generate icon 	You can click the icon to generate the test data in the CSV format. See Generating data from a Data definition on page 324 .
The Delete icon 	You can click the icon to delete the Data definition. See Deleting Data definitions on page 273 .
The Catalog tab	Provides a structured overview of the available built-in and custom generators for the selected project.
The History tab	<p>Provides an overview of all the previously generated test data. The following information about the history of the generated data is displayed:</p> <ul style="list-style-type: none"> ◦ The month, date, and year of the generated test data. ◦ The status of the download that indicates whether the data generation was successful or failed. ◦ The number of rows in the generated test data. ◦ The initials of the project member who downloaded the generated test data. <p>You can also use the following options to work with the past data:</p> <ul style="list-style-type: none"> ◦ You can click the preview icon  to review the already generated data before taking any further actions. ◦ You can click the generate icon  to regenerate the same test data, if required. See Generating data from history of a Data definition on page 326. ◦ You can click the delete icon  to delete specific test data history. You get a clutter-free view of the most relevant data and maintain your data generation history effectively.
The Details tab	<p>Provides following details about the Data definition and its associated Git details:</p> <ul style="list-style-type: none"> ◦ Name of the branch that contains the data definition. ◦ Description for the Data definition, if any. ◦ Git-related details for the Data definition as follows: <ul style="list-style-type: none"> ▪ Asset ID: The ID uniquely identifies the data definition. You also have the option to copy the ID for easy reference or use. ▪ Repo: The repository name and path where the Data definition is stored is displayed. ▪ Path: The exact location of the data definition within the repository is displayed.

Element	Description
	<ul style="list-style-type: none"> ▪ Commit: The Git commit message, the name of the member who performed the changes, and the date and time of the commit are displayed. ▪ The copy button: You can click the button to copy the complete SHA commit ID for the data definition. ▪ The commit ID button: You can click the button to view the changes that were made in the commit directly within Git. ▪ The browse button: You can click the button to view the specific location in the Git history where the commit was performed for the data definition.
The Data definition tabs	<p>When you open a Data definition, the name of the currently opened Data definition is displayed as a tab in the Data page. Beside this tab, additional tabs with the names of all other data definitions that are currently open on the Data page are also displayed. You can use these tabs to navigate between different Data definitions easily.</p> <p>Also, when you hover over the tab of a Data definition, a tooltip is displayed that provides the location and repository details where the Data definition is stored.</p>
The Branch list	The list displays the name of the branch where you currently opened the Data definition. You can also use the list to choose and select a different branch from the available options, if required. You can switch between branches easily.



Note: The **Save** icon  and the **Delete** icon  are displayed in the Data definition page only if you are in an Edit branch. If you are in a branch other than the Edit branch, then an **Edit** icon  is displayed in the Data definition. You can click the **Edit** icon  to perform any modification to the Data definition. Also, if you do not have either the owner or the tester role for the Data definition, you cannot save, modify, or delete the Data definition.

Results

You have viewed the contents of the Data definition in your project.

What to do next

You can perform any of the following tasks:

- Modify the Data definition. See [Modifying a Data definition on page 264](#).

Deleting Data definitions

You can delete Data definitions to remove them when the data definitions are outdated or no longer relevant to the current project. Additionally, when Data definitions contain errors or inaccuracies that can lead to misleading or incorrect fabricated data, then you might want to remove them. This ensures that the project remains clutter-free, optimizes the performance of HCL DevOps Test Hub (Test Hub), and reduces storage requirements.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a Data definition. See [Creating an empty Data definition on page 251](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Open the Data definition in the Edit branch.

5. Click the **Delete** icon to delete the Data definition.

The **Delete Data definition** dialog is displayed.

6. Click the **Delete** button.

Results

You have deleted a Data definition from your project.

What to do next

You can publish the changes performed to the data definition (that is, deletion of the Data definition) to the Git repository so that other members of the project do not see the deleted Data definition. See [Publishing changes in an Edit branch on page 267](#).

Basic generators

You can use the built-in basic generators in Data definitions with or without modifying the default configurations of the generators.

The Data fabrication feature of HCL DevOps Test Hub (Test Hub) contains a collection of built-in data generators available in the **Basic** category on the **Catalog** tab of a Data definition.

Basic generators are the built-in generators of the Data fabrication feature of Test Hub that you can use to generate diverse test data. These generators provide commonly used data types and patterns that you can select and configure in a Data definition to generate realistic test data.

The basic generators that you can use in Test Hub are as follows:

- Address
- Boolean
- Dataset
- DateTime
- Enumerator
- Function
- Name
- Number
- Phone
- Regex
- String
- Text
- UUID

Every basic generator has unique parameters with default features. When you click a basic generator that is added as a field in the Data definition, the corresponding unique parameters with their default configuration are displayed in a new dialog. The new dialog is named after the generator that you select.

You can change the default configurations of any basic generator only when it is added as a field in a Data definition. You can use the changed configurations for any basic generator in the Data definition by saving the Data definition. The changed configuration is applicable only in the Data definition that contains it and is not available for use from the **Catalog** tab.

If you want to save the modified generators, you can create them as custom generators and save them in a custom category in the **Catalog** tab.

You can find details about the parameters in each of the basic generators along with their default configuration and the configurations that you can change or modify. The parameters of the basic generators are as follows:

- [Parameters of the Address generator on page 275](#)
- [Parameters of the Boolean generator on page 279](#)
- [Parameters of the Dataset generator on page 283](#)
- [Parameters of the DateTime generator on page 285](#)
- [Parameters of the Enumerator generator on page 289](#)
- [Parameters of the Function generator on page 293](#)
- [Parameters of the Name generator on page 299](#)
- [Parameters of the Number generator on page 303](#)
- [Parameters of the Phone generator on page 307](#)

- [Parameters of the Regex generator on page 311](#)
- [Parameters of the String generator on page 314](#)
- [Parameters of the Text generator on page 319](#)
- [Parameters of the UUID generator on page 322](#)

Parameters of the Address generator

After you add the Address generator in a field of a Data definition, you can customize the existing configurations of the field.




When you click the Address generator of a specific field in the Data definition in the Edit branch, the unique parameters of the Address generator that contains the existing configuration details for that field are displayed in the **Address** dialog.





The following table provides the list of UI elements of the **Address** dialog with their description, default features, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Address .	The field is not editable.	NA
The Distribution field	Determines whether the addresses are randomly generated in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the addresses to follow a random distribution when generated in the test data. • Clear the Random option if you want the addresses to follow a fixed distribution and to be non-random. <p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribution for addresses using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains addresses with the postal codes and 70% to another configuration that contains</p>

UI element	Description	Default feature	Action
			<p>the address with no postal codes, then the address column of the generated data displays the values accordingly.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>
The Duplicates field	Determines whether duplicate or unique addresses are present in the generated data. Contains the Allowed option.	The Allowed option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated addresses to include duplicate entries. This means that some addresses might appear more than once in the generated data. • Clear the Allowed option if you want to ensure unique addresses with no duplicates. This will prevent any duplicate addresses from appearing in the generated data.
The Blank percentage field	Determines the frequency of blank addresses in the generated data.	Displays 0 and is editable.	Enter a number from 0 to 100. For example, if you enter 10, approximately 10% of the secondary addresses fields in the generated data will be empty or blank.
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.
The Percentage field	Determines the proportion of the generated addresses when you do not select the Random option.	Editable when Random is not selected.	<p>If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of addresses to be generated.</p> <p>For example, if you enter 50, the generated data will display the address column with a 50% occurrence rate.</p>

UI element	Description	Default feature	Action
The First Line option	Determines whether the generated addresses include the first line of the address.	The First Line option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated addresses to include the first line of the address. • Clear the Allowed option if you want to exclude the first line from the generated addresses.
The Second Line option	Determines whether the generated addresses include the second line of the address.	The Second Line option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated addresses to include the second line of the address. • Clear the Allowed option if you want to exclude the second line from the generated addresses.
The City option	Determines whether the generated addresses include the city name.	The City option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated addresses to include the name of the city. • Clear the Allowed option if you want to exclude the name of the city from the generated addresses.
The Zip Code option	Determines whether the generated addresses include the postal code.	The Zip Code option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated addresses to include the postal code. • Clear the Allowed option if you want to exclude the postal code from the generated addresses.

UI element	Description	Default feature	Action
The Country option	Determines whether the generated addresses include the country name.	The Country option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated addresses to include the name of the country. • Clear the Allowed option if you want to exclude the name of the country from the generated addresses.
The Add new configuration tab	You can add additional address configurations with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays <code>Configuration-1</code> and is editable. • The Percentage field is editable when the Random option is not selected. • The selected options from the previous configuration remain selected, and the unselected options remain unselected for the following elements: <ul style="list-style-type: none"> ◦ First Line ◦ Second Line ◦ City ◦ Zip Code ◦ Country 	<p>Click the tab to create additional address configurations.</p> <p>For example, you can have one configuration for addresses with the postal code, and another configuration for addresses with no postal code. You can similarly customize the values, distribution, and percentage as required.</p> <p> Note: You can click the delete icon  to delete any additional configuration that you do not need.</p>
The Save to catalog	After the customization of the Address generator is complete, you can use the Address generator as the	NA	Click the Save to catalog icon  to save the new generator with a unique name in the catalog.

UI element	Description	Default feature	Action
alog option	base generator to create a new custom generator.		You can reuse the same configuration at a later time in different scenarios without having to set it up again. See Creating custom generators and categories from Data definitions on page 331 .
The preview icon 	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.
The close icon 	NA	NA	Click the icon to exit the Address generator dialog.

Parameters of the Boolean generator




After you add the Boolean generator in a field of a Data definition, you can customize the existing configurations of the field.





When you click the Boolean generator of a specific field in the Data definition in the Edit branch, the unique parameters of the Boolean generator that contains the existing configuration details for that field are displayed in the **Boolean** dialog.

The following table provides the list of UI elements of the **Boolean** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Boolean .	The name is not editable.	NA

UI element	Description	Default feature	Action
The Distribution field	Determines if the Boolean values are randomly generated or not in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the Boolean values, when generated in the test data, to follow a random distribution. Clear the Random option if you want the Boolean to follow a fixed distribution and to be non-random. <p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribution for the Boolean values using the Percentage field.</p> <p>For example, if you set the percentage distribution of 100% and provide a value as <code>True</code>, then all the generated Boolean values in the corresponding column will be true.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>
The Blank percentage field	Determines the frequency of blank values in the corresponding column of the generated data.	Displays 0 and is editable.	<p>Enter a number from 0 to 100.</p> <p>For example, if you enter 10, approximately 10% of the Boolean values in the corresponding column of the generated data will be blank.</p>
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.
The Percentage	Determines the proportion of the generated Boolean values	Editable when Random is not selected.	If you did not select the Random option for distribution, enter a number from 0 to 100 in the

UI element	Description	Default feature	Action
age field	when you do not select the Random option.		<p>Percentage field to set the percentage of the Boolean values to be generated.</p> <p>For example, if you enter 50, the generated data will display the Boolean column with a 50% occurrence rate.</p>
The Format section	Determines the additional details about the format of the generated Boolean values for the selected configuration.	NA	Provide information about the format of the Boolean values for the selected configuration. For more details about the section, see The Format section on page 282 .
The Add new configuration tab	You can add additional configurations for boolean values with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays <code>Con-figuration-1</code> and is editable. • The Percentage field is editable when the Random option is not selected. • The Value and Distribution fields display the same values that were provided in the previous configuration. 	<p>Click the tab to create additional configurations for Boolean values. This is useful when you need different sets of Boolean values in your synthetic data.</p> <p>For example, you can have one configuration for <code>Is married</code> with a fixed distribution, and another configuration for <code>Has children</code> with a random distribution. You can similarly customize the values, distribution, and percentage as required.</p> <p> Note: You can click the delete icon  to delete any additional configuration that you do not need.</p>
The Save to catalog option	After the customization of the Boolean generator is complete, you can use the Boolean generator as the base generator to create a new custom generator.	NA	<p>Click the Save to catalog icon .</p> <p>You can reuse the same configuration at a later time in different scenarios without having to set it up again from scratch. See Creating custom generators and categories from Data definitions on page 331.</p>

UI element	Description	Default feature	Action
The preview icon 	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.
The close icon 	NA	NA	Click the icon to exit the Boolean generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **Dataset** generator should be equal to 100.

The Format section

You can add additional details about the format of the generated Boolean values for a selected configuration in the **Format** section of the **Boolean** dialog. The following table provides the list of UI elements of the section with their description, default features, and actions that you can perform in the section to customize the configuration:

UI element	Description	Default feature	Action
The Value fields	Contains two editable fields.	Displays Yes in the first field and No in the second field.	You can modify the default values to any other suitable Boolean values.
The Distribution fields	Contains two editable fields. The field determines the proportion of the entered values when you do not select the Random option.	Editable when Random is not selected. Displays 60% and 40% as default values.	If you did not select the Random option, enter a number from 0 to 100 in the Distribution field for both values. Note: The total percentage of the distribution corresponding to both values must be equal to 100.

UI element	Description	Default feature	Action
The Random option	Determines if the Boolean values are randomly generated or not.	The option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the Boolean values, when generated in the test data, to follow a random distribution. Clear the Random option if you want the Boolean values to follow a fixed distribution and to be non-random.




Parameters of the Dataset generator





After you add the Dataset generator in a field of a Data definition, you can customize the existing configurations of the field.

When you click the Dataset generator of a specific field in the Data definition in the Edit branch, the unique parameters of the Dataset generator that contains the existing configuration details for that field are displayed in the **Dataset** dialog.

The following table provides the list of UI elements of the **Dataset** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Dataset .	The name is not editable.	NA
The Blank percentage field	Determines the frequency of blank datasets in the generated data.	Displays 0 and is editable.	<p>Enter a number from 0 to 100.</p> <p>For example, if you enter 10, approximately 10% of the fields corresponding to datasets in the generated data will be blank.</p>
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.

UI element	Description	Default feature	Action
The Percentage field	Determines the proportion of the generated rows with datasets.	Editable.	Enter a number from 0 to 100 in the field to set the percentage of rows with datasets to be generated. For example, if you enter 50, the generated data will display the datasets column with a 50% occurrence rate.
The Dataset field	Allows you to browse and select a dataset file from the system.	NA	Click the Browse option to select a dataset file from your system. The selected dataset value is displayed in the Dataset field.
The Column list	Allows you to select the desired column from the dataset for data generation.	NA	Click the list to select the column from the dataset.
The Add new configuration tab	You can add additional dataset configurations with specific dataset and column selections, if required.	<ul style="list-style-type: none"> • The Name field displays <code>Configuration-1</code> and is editable. • The Percentage field is editable. • The Dataset and Column fields display the same values that were provided in the previous configuration. 	Click the tab to create additional dataset configurations, and then add the required components for the additional dataset configuration. For example, if the previous configuration included a dataset with the Address column, you can add a new dataset in another configuration to include another dataset with the Email column.  Note: You can click the delete icon  to delete any additional configuration that you do not need.
The Save to catalog option	After the customization of the Dataset generator is complete, you can use the Dataset generator as the base generator to create a new custom generator.	NA	Click the Save to catalog icon  . You can reuse the same configuration at a later time in different scenarios without having to set it up again from scratch. See

UI element	Description	Default feature	Action
			Creating custom generators and categories from Data definitions on page 331.
The preview icon 	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.
The close icon 	NA	NA	Click the icon to exit the Dataset generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **Dataset** generator should be equal to 100.

Parameters of the DateTime generator

After you add the DateTime generator in a field of a Data definition, you can customize the existing configurations of the field.




When you click the DateTime generator of a specific field in the Data definition in the Edit branch, the unique parameters of the DateTime generator that contains the existing configuration details for that field are displayed in the **DateTime** dialog.


The following table provides the list of UI elements of the **DateTime** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, DateTime .	The name is not editable.	NA

UI element	Description	Default feature	Action
The Dis-tribution field	Determines if the date and time are randomly generated or not in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the date and time, when generated in the test data, to follow a random distribution. • Clear the Random option if you want the date and time to follow a fixed distribution and to be non-random. <p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribution for the date and time using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains the dates of weekdays with a fixed time range and 70% to another configuration that contains the dates of weekends with a different time range, then the enumerator column of the generated data displays the values accordingly.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>
The Du-plicates field	Determines if duplicate or unique date and time are present in the generated data. Contains the Allowed option.	The Allowed option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated date and time to include duplicate entries. This means that some of the date and time values might appear more than once in the generated data. • Clear the Allowed option if you want to ensure a unique date and time with no duplicates. This will prevent any duplicate date and time from appearing in the generated data.

UI element	Description	Default feature	Action
The Blank percentage field	Determines the frequency of blank date and time values in the generated data.	Displays 0 and is editable.	Enter a number from 0 to 100. For example, if you enter 10, approximately 10% of the date and time values in the generated data will be blank.
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.
The Percentage field	Determines the proportion of the generated date and time values when you do not select the Random option.	Editable when Random is not selected.	<p>If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of the date and time values to be generated.</p> <p>For example, if you enter 50, the generated data will display the column corresponding to the date and time values with a 50% occurrence rate.</p>
The Pattern list	Allows you to select the date pattern for data generation.	The default pattern is mm/dd/yyyy .	Select the required date pattern from the list of available options.
The Limit dates section	Allows you to define limitations on the generated dates.	NA	<ul style="list-style-type: none"> • Either enter the dates in the From and To fields in the format of dd/mm/yyyy or select the dates from the corresponding calendar icons to reflect in the fields. • Select the days of the week (S, M, T, W, T, F, S) in the Exclude section to exclude those days from the From and To date fields. • Enter multiple dates in the Specific dates to exclude field, separated by commas, to exclude specific dates from the generation.
The Limit times	Allows you to define limitations on the generated times.	NA	Enter the times in the From and To fields in the AM and PM format.

UI element	Description	Default feature	Action
section			
The Add new configuration tab	You can add additional date and time configurations with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays <code>Configuration-1</code> and is editable. • The Percentage field is editable when the Random option is not selected. • The following sections display the same values that were provided in the previous configuration: <ul style="list-style-type: none"> ◦ Limit dates ◦ Limit times 	<p>Click the tab and then add the required components for the additional date and time configuration.</p> <p>For example, you can have one configuration for weekdays with a fixed time range, and another configuration for weekends with a different time range.</p> <p> Note:</p> <p>You can click the delete icon  to delete any additional configuration that you do not need.</p>
The Save	After the customization of the DateTime generator is	NA	Click the Save to catalog icon  .

UI element	Description	Default feature	Action
to catalog option	complete, you can use the DateTime generator as the base generator to create a new custom generator.		You can reuse the same configuration at a later time in different scenarios without having to set it up again from scratch. See Creating custom generators and categories from Data definitions on page 331 .
The close icon 	NA	NA	Click the icon to exit the DateTime generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **DateTime** generator should be equal to 100.

Parameters of the Enumerator generator




After you add the Enumerator generator in a field of a Data definition, you can customize the existing configurations of the field.


When you click the Enumerator generator of a specific field in the Data definition in the Edit branch, the unique parameters of the Enumerator generator that contains the existing configuration details for that field are displayed in the **Enumerator** dialog.

The following table provides the list of UI elements of the **Enumerator** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Enumerator .	The name is not editable.	NA

UI element	Description	Default feature	Action
The Distribution field	Determines if the enumerator values are randomly generated or not in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the enumerator values, when generated in the test data, to follow a random distribution. • Clear the Random option if you want the enumerator values to follow a fixed distribution and to be non-random. <p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribution for the enumerator values using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains the names of colors and 70% to another configuration that contains the names of shapes, then the enumerator column of the generated data displays the values accordingly.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>
The Blank percentage field	Determines the frequency of blank enumerator values in the generated data.	Displays 0 and is editable.	<p>Enter a number from 0 to 100.</p> <p>For example, if you enter 10, approximately 10% of the enumerator values in the generated data will be blank.</p>
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.

UI element	Description	Default feature	Action
The Percentage field	Determines the proportion of the generated enumerated values when you do not select the Random option.	Editable when Random is not selected.	<p>If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of enumerator values to be generated.</p> <p>For example, if you enter 50, the generated data will display the enumerator column with a 50% occurrence rate.</p>
The Format section	Determines the additional details about the format of the generated enumerator values for the selected configuration.	NA	Provide information about the format of the enumerator values for the selected configuration. For more details about the section, see The Format section on page 292 .
The Add new configuration tab	You can add additional enumerator value configurations with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays Configuration-1 and is editable. • The Percentage field is editable when the Random option is not selected. • The Value and Distribution fields display the same values that were provided in the previous configuration. 	<p>Click the tab and then add the required components for the additional configuration.</p> <p>For example, you can have one configuration for colors with a fixed distribution, and another configuration for shapes with a random distribution. You can similarly customize the values, distribution, and percentage as required.</p> <p> Note: You can click the delete icon  to delete any additional configuration that you do not need.</p>
The Save	After the customization of the Enumerator generator is com-	NA	Click the Save to catalog icon  .


UI element	Description	Default feature	Action
to complete, you can use the Enumerator generator as the base generator to create a new custom generator.			You can reuse the same configuration at a later time in different scenarios without having to set it up again from scratch. See Creating custom generators and categories from Data definitions on page 331 .
The close icon 	NA	NA	Click the icon to exit the Enumerator generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **Enumerator** generator should be equal to 100.

The Format section

You can add additional details about the format of the generated enumerator values for a selected configuration in the **Format** section of the **Enumerator** dialog. The following table provides the list of UI elements of the section with their description, default features, and actions that you can perform in the section to customize the configuration:

UI element	Description	Default feature	Action
The Value fields	You can add multiple editable fields and provide desired values.	Displays three editable fields with RED , GREEN , and BLUE as the corresponding values.	You can modify the default values to any other suitable enumerator values or add additional values.
The Distribution fields	You can add multiple editable fields. The field determines the proportion of the entered values when you do not select the Random option.	Editable when Random is not selected. Displays 20%, 30%, and 50% as default values.	If you did not select the Random option, enter a number from 0 to 100 in the Distribution fields for all the values.  Note: The total percentage of distribution corresponding to both values must be equal to 100.

UI element	Description	Default feature	Action
The Random option	Determines if the enumerator values are randomly generated or not.	The option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the enumerator values, when generated in the test data, to follow a random distribution. Clear the Random option if you want the enumerator values to follow a fixed distribution and to be non-random.

Parameters of the Function generator

After you add the Function generator in a field of a Data definition, you can customize the existing configurations of the field.








When you click the Function generator of a specific field in the Data definition in the Edit branch, the unique parameters of the Function generator that contains the existing configuration details for that field are displayed in the **Function** dialog.

The following table provides the list of UI elements of the **Function** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Function .	The name is not editable.	NA
The Distribution field	Determines if the values are randomly generated or not in the function column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the values, when generated in the function column, to follow a random distribution. Clear the Random option if you want the function values to follow a fixed distribution and to be non-random.

UI element	Description	Default feature	Action
			<p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribution for the Function values using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains the names in uppercase and 70% to another configuration that contains the names in lowercase, then the Function column of the generated data displays the values accordingly.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>
The Duplicates field	<p>Determines if duplicate or unique function values are present in the generated data. Contains the Allowed option.</p>	The Allowed option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated function values to include duplicate entries. • Clear the Allowed option if you want to ensure unique function values with no duplicates. This will prevent any duplicate values from appearing in the function column.
The Blank percentage field	Determines the frequency of blank function values in the generated data.	Displays 0 and is editable.	Enter a number from 0 to 100. For example, if you enter 10, approximately 10% of the function values in the generated data will be blank.
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.

UI element	Description	Default feature	Action
The Percentage field	Determines the proportion of the generated function values when you do not select the Random option.	Editable when Random is not selected.	If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of function values to be generated. For example, if you enter 50, the generated data will display the function column with a 50% occurrence rate.
The Function section	Determines the additional details about the function for the selected configuration.	NA	Provide information about the function for the selected configuration. For more details about the section, see The Function section on page 297 .
The Add new configuration tab	You can add additional function configurations with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays configuration-1 and is editable. 	<p>Click the tab and then add the required components for the additional configuration.</p> <p>For example, you can have one configuration for uppercase names with a 30% distribution and another configuration for lowercase names with a 70% distribution. You can similarly customize the values, distribution, and percentage as required.</p>

UI element	Description	Default feature	Action
		<ul style="list-style-type: none"> • The Percentage field is editable when the Random option is not selected. • The Function field displays the same values that were provided in the previous configuration. 	 Note: You can click the delete icon  to delete any additional configuration that you do not need.
The Save to catalog option	After the customization of the Function generator is complete, you can use the Function generator as the base generator to create a new custom generator.	NA	Click the Save to catalog icon  . You can reuse the same configuration at a later time in different scenarios without having to set it up again from scratch. See Creating custom generators and categories from Data definitions on page 331 .
The preview icon 	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.
The close icon 	NA	NA	Click the icon to exit the Function generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **Function** generator should be equal to 100.

The Function section

You can add additional details about the function for a selected configuration in the **Function** section of the **Function** dialog. The following table provides the list of UI elements of the section with their description, default features, and actions that you can perform in the section to customize the configuration:

UI element	Description	Default feature	Action
The Function field	Combines the built-in methods and the Data definition fields to create custom functions.	Empty	You can either enter a function or select values from the Fields and Methods scroll areas to create a function.
The Items for function field	Finds matching results in both the Methods and Fields scroll areas based on the entered text.	NA	You can type a few letters in the Items for function field, and matching results from both the Methods and Fields scroll areas will be displayed.
The Methods scroll area	Displays predefined Java methods that you can use to perform specific operations on entity fields within the function.	NA	<p>When you double-click on an available method, the corresponding parameter is added to the Function field.</p> <p>When you hover over a method in the scroll area, the corresponding function, description, and example are displayed.</p> <p>The Data fabrication feature of HCL DevOps Test Hub (Test Hub) supports a variety of built-in methods. See The supported built-in methods on page 298.</p>
The Fields scroll area	Displays the names of all the fields used in the Data definition. You can double-click a field to add it to the Function field and incorporate it into the custom function.	NA	When you double-click an item in the Fields scroll area, the corresponding parameter is added to the Function field.

The supported built-in methods

The Data fabrication feature of Test Hub supports a variety of built-in methods. The following table provides the list of built-in methods, their description, and relevant examples that you can use as functions:

Method	Description	Example
<code>replace(oldString, newString)</code>	Replaces all occurrences of the old string with the new string	<code>F\$address.replace('road', 'street')</code>
<code>format(formatString, args)</code>	Returns a formatted string using format string and arguments	<code>String.format('Hello %s', F\$name)</code>
<code>split(matchString)</code>	Splits the string around the given matching string	<code>F\$name.split(" ")[0].concat('@gmail.com')</code> . In this example, the name is split where a white space is present and only the first part of the split is considered. Also, the first part of the split is appended with <code>@gmail.com</code> . If the input name is <code>Amy Smith</code> , then the result of the function is <code>Amy@gmail.com</code> .
<code>join(delimiter, stringElements)</code>	Joins string elements by a delimiter.	<code>String.join(',', F\$id, F\$name)</code>
<code>length()</code>	Returns the length of the string	<code>F\$name.length</code>
<code>isBlank()</code>	Returns True only if the string has zero length or contains only white spaces	<code>F\$address.isBlank</code>
<code>trim()</code>	Removes leading and trailing spaces of the string	<code>F\$phone.trim</code>
<code>toUpperCase()</code>	Converts all characters into uppercase	<code>F\$name.toUpperCase</code>
<code>toLowerCase()</code>	Converts all characters into lowercase	<code>F\$name.toLowerCase</code>
<code>concat</code>	Concatenates the specified string to	<code>'Hello '.concat(F\$name)</code>

Method	Description	Example
	the end of the selected string	

Parameters of the Name generator

After you add the Name generator in a field of a Data definition, you can customize the existing configurations of the field.








When you click the Name generator of a specific field in the Data definition in the edit branch, the unique parameters of the Name generator that contains the existing configuration details for that field are displayed in the **Name** dialog.

The following table provides the list of UI elements of the **Name** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Name .	The name is not editable.	NA
The Distribution field	Determines if the names are randomly generated or not in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the names, when generated in the test data, to follow a random distribution. Clear the Random option if you want the names to follow a fixed distribution and to be non-random, when you add a new configuration. <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>

UI element	Description	Default feature	Action
The Duplicates field	Determines if duplicate or unique names are present in the generated data. Contains the Allowed option.	The Allowed option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the generated names to include duplicate entries. Clear the Allowed option if you want to ensure unique names with no duplicates. This will prevent any duplicate names from appearing in the generated data. <p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribution for the names using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains the names with titles and 70% to another configuration that does not contain the title, then the names column of the generated data displays the values accordingly.</p>
The Blank percentage field	Determines the frequency of blank names in the generated data.	Displays 0 and is editable.	Enter a number from 0 to 100. For example, if you enter 10, approximately 10% of the names in the generated data will be blank.
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.
The Percentage field	Determines the proportion of the generated names when you do not select the Random option.	Editable when Random is not selected.	If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of names to be generated.

UI element	Description	Default feature	Action
			For example, if you enter 50, the generated data will display the names column with a 50% occurrence rate.
The Title option	Determines whether the generated names include the title.	The Title option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated names to include the title. • Clear the Allowed option if you want to exclude the title from the generated names.
The First Name option	Determines whether the generated names include the first name.	The First Name option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated names to include the first name. • Clear the Allowed option if you want to exclude the first name from the generated names.
The Last Name option	Determines whether the generated names include the last name.	The Last Name option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated names to include the last name. • Clear the Allowed option if you want to exclude the last name from the generated names.
The Add new configuration tab	You can add additional name configurations with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays <i>Configuration-1</i> and is editable. • The Percentage field is editable when the Random 	<p>Click the tab and then add the required components for the additional configuration.</p> <p>For example, you can have one configuration for names with titles and another configuration for names without titles. You can similarly customize the values, distribution, and percentage as required.</p>

UI element	Description	Default feature	Action
		<p>option is not selected.</p> <ul style="list-style-type: none"> The selected options from the previous configuration remain selected, and the unselected options remain unselected for the following elements: <ul style="list-style-type: none"> Title First Name Last Name 	<p> Note: You can click the delete icon  to delete any additional configuration that you do not need.</p>
The Save to catalog option	After the customization of the Name generator is complete, you can use the Name generator as the base generator to create a new custom generator.	NA	<p>Click the Save to catalog icon .</p> <p>You can reuse the same configuration at a later time in different scenarios without having to set it up again from the scratch. See Creating custom generators and categories from Data definitions on page 331.</p>
The preview icon 	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.
The close icon 	NA	NA	Click the icon to exit the Name generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **Name** generator should be equal to 100.

Parameters of the Number generator

After you add the Number generator in a field of a Data definition, you can customize the existing configurations of the field.







When you click the Number generator of a specific field in the Data definition in the Edit branch, the unique parameters of the Number generator that contains the existing configuration details for that field are displayed in the **Number** dialog.


The following table provides the list of UI elements of the **Number** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Number .	The name is not editable.	NA
The Distribution field	Determines if the numbers are randomly generated or not in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the numbers, when generated in the test data, to follow a random distribution. Clear the Random option if you want the numbers to follow a fixed distribution and to be non-random. <p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribution for the numbers using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains numbers with 2 decimal values and 70% to another configuration that contains numbers with no decimal val-</p>

UI element	Description	Default feature	Action
			<p>ues, then the number column of the generated data displays the values accordingly.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>
The Duplicates field	Determines if duplicate or unique numbers are present in the generated data. Contains the Allowed option.	The Allowed option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated numbers to include duplicate entries. • Clear the Allowed option if you want to ensure unique numbers with no duplicates. This will prevent any duplicate numbers from appearing in the generated data.
The Blank percentage field	Determines the frequency of blank numbers in the generated data.	Displays 0 and is editable.	Enter a number from 0 to 100. For example, if you enter 10, approximately 10% of the numbers in the generated data will be blank.
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.
The Percentage field	Determines the proportion of the generated numbers when you do not select the Random option.	Editable when Random is not selected.	<p>If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of numbers to be generated.</p> <p>For example, if you enter 50, the generated data will display the number column with a 50% occurrence rate.</p>

UI element	Description	Default feature	Action
The Numbers section	Determines either the range of numbers or the number of digits that appear in the generated data. You can select either the Range or # of Digits button and then enter the corresponding values in the Min and Max fields.	The # of Digits button is selected with a minimum digit of 1 and a maximum digit of 10 .	<ul style="list-style-type: none"> • Enter numeric values in the Min and Max fields to set the minimum and maximum number of digits respectively. The # of Digits button is selected as the default feature. • Click the Range button to switch to range-based number generation and then enter numeric values in the Min and Max fields to set the minimum and maximum range of numbers.
The Decimal places section	Determines the range of decimal places that are displayed in the generated numbers.	The Min and Max fields in the section are blank.	Enter numeric values in the Min and Max fields to set the minimum and maximum number of decimal places, respectively, in the generated numbers.
The Add new configuration tab	You can add additional configurations for numbers with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays <code>Con-figuration-1</code> and is editable. 	<p>Click the tab and then add the required components for the additional configuration.</p> <p>For example, you can have one configuration for numbers with fixed decimal places and another configuration with no decimal places. You can similarly customize the values, distribution, and percentage as required.</p>

UI element	Description	Default feature	Action
		<ul style="list-style-type: none"> • The Percentage field is editable when the Random option is not selected. • The fields or options in the following sections display the same values that were provided in the previous configuration: <ul style="list-style-type: none"> ◦ Numbers ◦ Decimal places 	<p> Note: You can click the delete icon  to delete any additional configuration that you do not need.</p>
The Save to catalog option	After the customization of the Number generator is complete, you can use the Number generator as the base generator to create a new custom generator.	NA	<p>Click the Save to catalog icon .</p> <p>You can reuse the same configuration at a later time in different scenarios without having to set it up again from the scratch. See Creating custom generators and categories from Data definitions on page 331.</p>
The preview icon 	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.

UI element	Description	Default feature	Action
The close icon 	NA	NA	Click the icon to exit the Number generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **Number** generator should be equal to 100.

Parameters of the Phone generator

After you add the Phone generator in a field of a Data definition, you can customize the existing configurations of the field.

When you click the Phone generator of a specific field in the Data definition in the edit branch, the unique parameters of the Phone generator that contains the existing configuration details for that field are displayed in the **Phone** dialog.



The following table provides the list of UI elements of the **Phone** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Phone .	The name is not editable.	NA
The Distribution field	Determines if the phone numbers are randomly generated in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the phone numbers, when generated in the test data, to follow a random distribution. Clear the Random option if you want the phone numbers to follow a fixed distribution and to be non-random. <p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribu-</p>

UI element	Description	Default feature	Action
			<p>tion for the phone numbers using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains a specific pattern with the UK as the country and 70% to another configuration that contains another pattern with France as the country, then the column of the phone number of the generated data displays the values accordingly.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>
The Duplicates field	Determines if duplicate or unique phone numbers are present in the generated data. Contains the Allowed option.	The Allowed option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated phone numbers to include duplicate entries. • Clear the Allowed option if you want to ensure unique phone numbers with no duplicates. This will prevent any duplicate phone numbers from appearing in the generated data.
The Blank percentage field	Determines the frequency of blank phone numbers in the generated data.	Displays 0 and is editable.	<p>Enter a number from 0 to 100.</p> <p>For example, if you enter 10, approximately 10% of the phone numbers in the generated data will be blank.</p>
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.

UI element	Description	Default feature	Action
The Percentage field	Determines the proportion of the generated phone numbers when you do not select the Random option.	Editable when Random is not selected.	<p>If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of phone numbers to be generated.</p> <p>For example, if you enter 50, the generated data will display the phone numbers column with a 50% occurrence rate.</p>
The Format list	Determines the desired format for the generated phone numbers.	NA	Select a specific format from the list.
The Country list	Determines the country for which you want to generate phone numbers. <Not sure why this list is provided. Selecting or not selecting from this list has no changes in the generated data - Need to confirm with the team>	NA	Select a specific country name from the list.
The Add new configuration tab	You can add additional phone number configurations with specific components and settings, if required.	<ul style="list-style-type: none"> The Name field displays Configuration-1 	<p>Click the tab and then add the required components for the additional configuration.</p> <p>For example, you can have one configuration for colors with a fixed distribution, and another configuration for shapes with a random distribution. You can similarly customize the values, distribution, and percentage as required.</p>

UI element	Description	Default feature	Action
		<p>and is editable.</p> <ul style="list-style-type: none"> • The Percentage field is editable when the Random option is not selected. • The Format and Country fields display the same values that were provided in the previous configuration. 	<p> Note: You can click the delete icon  to delete any additional configuration that you do not need.</p>
The Save to catalog option	After the customization of the Phone generator is complete, you can use the Phone generator as the base generator to create a new custom generator.	NA	<p>Click the Save to catalog icon .</p> <p>You can reuse the same configuration at a later time in different scenarios without having to set it up again from the scratch. See Creating custom generators and categories from Data definitions on page 331.</p>
The preview	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.

UI element	Description	Default feature	Action
icon 			
The close icon 	NA	NA	Click the icon to exit the Phone generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **Phone** generator should be equal to 100.

Parameters of the Regex generator


After you add the Regex generator in a field of a Data definition, you can customize the existing configurations of the field.






When you click the Regex generator of a specific field in the Data definition in the edit branch, the unique parameters of the Regex generator that contains the existing configuration details for that field are displayed in the **Regex** dialog.

The following table provides the list of UI elements of the **Regex** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Regex .	The name is not editable.	NA
The Distribution field	Determines if the Regex values are randomly generated in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the Regex values, when generated in the test data, to follow a random distribution. Clear the Random option if you want the Regex values to follow a fixed distribution and to be non-random. <p>When you clear the option, the Percentage fields of all the existing configurations become ed-</p>

UI element	Description	Default feature	Action
			<p>itable. You can set a fixed distribution for the Regex values using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains Regex values of ten numeric values and 70% to another configuration that contains Regex values of five alphabets, then the Regex column of the generated data displays the values accordingly.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>
The Duplicates field	Determines if duplicate or unique Regex values are present in the generated data. Contains the Allowed option.	The Allowed option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated Regex values to include duplicate entries. • Clear the Allowed option if you want to ensure unique Regex values with no duplicates. This will prevent any duplicate Regex values from appearing in the generated data.
The Blank percentage field	Determines the frequency of blank Regex values in the generated data.	Displays 0 and is editable.	<p>Enter a number from 0 to 100.</p> <p>For example, if you enter 10, approximately 10% of the Regex values in the generated data will be blank.</p>
The Name field	Displays the custom name of the selected configuration.	Displays <i>Default</i> and is editable.	Enter an identifiable name to the selected configuration, if required.
The Percentage field	Determines the proportion of the generated Regex values when you do not select the Random option.	Editable when Random is not selected.	If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of the Regex values to be generated.

UI element	Description	Default feature	Action
			For example, if you enter 50, the generated data will display the column of the Regex values with a 50% occurrence rate.
The Regex field	Determines the custom regular expression pattern to generate data that matches the given pattern.	Displays [a-z]{4} .	<ul style="list-style-type: none"> • Edit the default Regex pattern and enter the desired custom Regex pattern. • Perform the following actions to use the helper, if you are not sure of the required Regex format: <ul style="list-style-type: none"> ◦ Click the Helper option. <p>The Regex helper dialog appears.</p> <ul style="list-style-type: none"> ◦ Enter the sample texts in the fields provided, and then click the Create regex option. <p>The Regex format is displayed in the Regex generated field.</p> <ul style="list-style-type: none"> ◦ Click Save to close the dialog and display the Regex value in the Regex field of the generator dialog.
The Add new configuration tab	You can add additional configurations of the Regex values with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays <code>Con-figuration-1</code> and is editable. • The Percentage field is editable when the Random option is not selected. 	<p>Click the tab and then add the required components for the additional configuration.</p> <p>For example, you can have one configuration that contains ten numerals, and another configuration that contains five alphabets. You can similarly customize the values, distribution, and percentage as required.</p> <p>You can click the delete icon  to delete any additional configuration that you do not need.</p>

UI element	Description	Default feature	Action
		<ul style="list-style-type: none"> The Regex fields display the same values that were provided in the previous configuration. 	
The Save to catalog option	After the customization of the Regex generator is complete, you can use the Regex generator as the base generator to create a new custom generator.	NA	Click the Save to catalog icon  . You can reuse the same configuration at a later time in different scenarios without having to set it up again from the scratch. See Creating custom generators and categories from Data definitions on page 331 .
The preview icon 	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.
The close icon 	NA	NA	Click the icon to exit the Regex generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **Regex** generator should be equal to 100.

Parameters of the String generator

After you add the String generator in a field of a Data definition, you can customize the existing configurations of the field.






When you click the String generator of a specific field in the Data definition in the Edit branch, the unique parameters of the String generator that contains the existing configuration details for that field are displayed in the **String** dialog.



The following table provides the list of UI elements of the **String** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, String .	The name is not editable.	NA
The Distribution field	Determines if the strings are randomly generated or not in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the strings, when generated in the test data, to follow a random distribution. • Clear the Random option if you want the strings to follow a fixed distribution and to be non-random. <p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribution for the strings using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains strings with special characters and 70% to another configuration that contains strings with no special characters, then the string column of the generated data displays the values accordingly.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>

UI element	Description	Default feature	Action
The Duplicates field	Determines if duplicate or unique strings are present in the generated data. Contains the Allowed option.	The Allowed option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the generated strings to include duplicate entries. Clear the Allowed option if you want to ensure unique strings with no duplicates. This will prevent any duplicate strings from appearing in the generated data.
The Blank percentage field	Determines the frequency of blank strings in the generated data.	Displays 0 and is editable.	<p>Enter a number from 0 to 100.</p> <p>For example, if you enter 10, approximately 10% of the strings in the generated data will be blank.</p>
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.
The Percentage field	Determines the proportion of the generated strings when you do not select the Random option.	Editable when Random is not selected.	<p>If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of strings to be generated.</p> <p>For example, if you enter 50, the generated data will display the string column with a 50% occurrence rate.</p>
The Minimum characters field	Determines the minimum number of digits for the string in the generated data.	Displays 10.	Edit the default number and enter the custom minimum value, if required.
The Maximum characters field	Determines the maximum number of digits for the string in the generated data.	Displays 1,000.	Edit the default number and enter the custom maximum value, if required.

UI element	Description	Default feature	Action
ters field			
The In-clude special characters option	Determines whether special characters must be included in the string.	The option is not selected.	Select the option if you want to include special characters in the string.
The In-clude digits option	Determines whether digits must be included in the string.	The option is not selected.	Select the option if you want to include digits in the string.
The Add new configuration tab	You can add additional string configurations with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays <code>Con-figuration-1</code> and is editable. • The Percentage field is editable when the 	<p>Click the tab and then add the required components for the additional configuration.</p> <p>For example, you can have one configuration for strings with special characters and another configuration for strings with no special characters. You can similarly customize the values, distribution, and percentage as required.</p>

UI element	Description	Default feature	Action
		<p>Random option is not selected.</p> <ul style="list-style-type: none"> • The fields or options in the following sections display the same values that were provided in the previous configuration: <ul style="list-style-type: none"> ◦ Minimum characters ◦ Maximum characters ◦ Include special characters ◦ Include digits 	<p> Note: You can click the delete icon  to delete any additional configuration that you do not need.</p>
The Save to catalog option	After the customization of the String generator is complete, you can use the String generator as the base generator to create a new custom generator.	NA	<p>Click the Save to catalog icon .</p> <p>You can reuse the same configuration at a later time in different scenarios without having to set it up again from the scratch. See Creating custom generators and categories from Data definitions on page 331.</p>
The preview	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.

UI element	Description	Default feature	Action
icon 			
The close icon 	NA	NA	Click the icon to exit the String generator dialog.



Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **String** generator should be equal to 100.

Parameters of the Text generator

After you add the Text generator in a field of a Data definition, you can customize the existing configurations of the field.








When you click the Text generator of a specific field in the Data definition in the edit branch, the unique parameters of the Text generator that contains the existing configuration details for that field are displayed in the **Text** dialog.

The following table provides the list of UI elements of the **Text** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, Text .	The name is not editable.	NA
The Distribution field	Determines if the texts are randomly generated or not in the corresponding column of the test data. Contains the Random option.	The Random option is selected.	<ul style="list-style-type: none"> Do not perform any action if you want the texts, when generated in the test data, to follow a random distribution. Clear the Random option if you want the texts to follow a fixed distribution and to be non-random.

UI element	Description	Default feature	Action
			<p>When you clear the option, the Percentage fields of all the existing configurations become editable. You can set a fixed distribution for the enumerator values using the Percentage field.</p> <p>For example, if you set the percentage distribution of 30% to a configuration that contains words with a count of 10 and 70% to another configuration that contains words with a count of 5, then the text column of the generated data displays the values accordingly.</p> <p>The percentage for each configuration is automatically rebalanced preserving the allocation in the existing configurations, when you add a configuration subsequently.</p>
The Duplicates field	Determines if duplicate or unique texts are present in the generated data. Contains the Allowed option.	The Allowed option is selected.	<ul style="list-style-type: none"> • Do not perform any action if you want the generated texts to include duplicate entries. • Clear the Allowed option if you want to ensure unique texts with no duplicates. This will prevent any duplicate texts from appearing in the generated data.
The Blank percentage field	Determines the frequency of blank texts in the generated data.	Displays 0 and is editable.	<p>Enter a number from 0 to 100.</p> <p>For example, if you enter 10, approximately 10% of the texts in the generated data will be blank.</p>
The Name field	Displays the custom name of the selected configuration.	Displays <code>Default</code> and is editable.	Enter an identifiable name to the selected configuration, if required.

UI element	Description	Default feature	Action
The Percentage field	Determines the proportion of the generated texts when you do not select the Random option.	Editable when Random is not selected.	If you did not select the Random option for distribution, enter a number from 0 to 100 in the Percentage field to set the percentage of texts to be generated. For example, if you enter 50, the generated data will display the text column with a 50% occurrence rate.
The Text type list	Determines the desired text type for the generated texts.	The Words option is selected.	Select a specific text type from the list. The available options are as follows: <ul style="list-style-type: none"> • Words • Sentences • Paragraphs
The Count field	Determines the count with respect to the selected text type.	Displays 10.	Edit the default count to enter the custom count of the selected text type.
The Add new configuration tab	You can add additional text configurations with specific components and settings, if required.	<ul style="list-style-type: none"> • The Name field displays <code>Configuration-1</code> and is editable. • The Percentage field is 	Click the tab and then add the required components for the additional configuration. For example, you can have one configuration for words with a count of 10, and another configuration for words with a count of 5. You can similarly customize the values, distribution, and percentage as required.

UI element	Description	Default feature	Action
		<p>editable when the Random option is not selected.</p> <ul style="list-style-type: none"> The Text type and Count lists display the same values that were provided in the previous configuration. 	<p> Note: You can click the delete icon  to delete any additional configuration that you do not need.</p>
The Save to catalog option	After the customization of the Text generator is complete, you can use the Text generator as the base generator to create a new custom generator.	NA	<p>Click the Save to catalog icon .</p> <p>You can reuse the same configuration at a later time in different scenarios without having to set it up again from the scratch. See Creating custom generators and categories from Data definitions on page 331.</p>
The preview icon 	The preview icon  is available for each configuration of a generator.	NA	Click the preview icon  to view the sample test data for the selected configuration of a generator.
The close icon 	NA	NA	Click the icon to exit the Text generator dialog.





Note: The total percentage displayed in the **Percentage** fields of all the configurations in the **Text** generator should be equal to 100.

Parameters of the UUID generator

After you add the UUID (Universal Unique Identifier) generator in a field of a data definition, you can customize the existing configurations of the field.

When you click the UUID generator of a specific field in the Data definition in the edit branch, the unique parameters of the UUID generator that contains the existing configuration details for that field are displayed in the **UUID** dialog.

The following table provides the list of UI elements of the **UUID** dialog with their default parameters, and the actions that you can perform in the dialog to customize the parameters for the selected field:

UI element	Description	Default feature	Action
The Generator field	Displays the name of the selected generator, that is, UUID .	The name is not editable.	NA
The Blank percentage field	Determines the frequency of the unique identifiers in the generated data.	Displays 0 and is editable.	Enter a number from 0 to 100. For example, if you enter 10, approximately 10% of the unique identifiers in the generated data will be blank.
The Save to catalog option	After the customization of the UUID generator is complete, you can use the UUID generator as the base generator to create a new custom generator.	NA	Click the Save to catalog icon  . You can reuse the same configuration at a later time in different scenarios without having to set it up again from the scratch. See Creating custom generators and categories from Data definitions on page 331 .
The close icon 	NA	NA	Click the icon to exit the UUID generator dialog.

Generation of test data

You can use the data fabrication feature in HCL DevOps Test Hub (Test Hub) to create realistic datasets for comprehensive testing scenarios. You can generate the test data as the Comma-Separated Values (CSV) files that is saved in your system. You can then use these files to replace any existing datasets in a test during the run time. If a dataset is already associated with a test, you can replace the dataset at runtime with the dataset that is created from the data fabrication feature of Test Hub. This allows you to integrate the generated test data into your test execution process, and you can ensure that your tests are performed with the secure and realistic data.

You can use the data fabrication feature of Test Hub to generate the test data in the following two ways:

- Generation of data from a Data definition
- Generation of data from history of a Data definition

Generation of data from a Data definition

Data definitions are fundamental building blocks that are used to define data structures and generate test data. You can create Data definitions and specify the characteristics of the data that you need for your testing scenarios. You can choose from the built-in data generators, configure new generators, define rules, and set restrictions and distribution for the Data definitions. You can then use the Data definitions to generate synthetic test data that closely resembles real-world scenarios.

You can use customized Data definitions to generate diverse test data that closely resembles real-world scenarios. You can preview a test data to ensure whether the data is in the expected format. You can have a glimpse of the generated test data that can help you to identify any potential issues, inconsistencies, or errors in the Data definitions. If you modify a Data definition but not save it, you can still preview the corresponding test data. However, you can generate the test data only after you save the Data definition. You can add multiple fields in a Data definition without associating them with generators and save the Data definition. However, you cannot preview or generate test data for such a Data definition.



Note: While configuring the generator parameters for a field in a Data definition, you must use the **Allowed** option of the **Duplicates** field of the generator with caution, particularly when you configure the data generator with a narrow criteria for data generation. If you enable the **Allowed** option in scenarios where duplications are not feasible within the defined criteria, then the preview or generation of test data might not be successful.

Also, the data distribution in a generated test data closely aligns with the configured weightage that you provide in the **Distribution** field of the generator. Therefore, the accuracy of the test data improves as more number of records are generated. See [Basic generators on page 273](#).

See [Generating data from a Data definition on page 324](#).

Generation of data from history of a Data definition

When you open a Data definition in Test Hub, you can view the history of the Data definition. You can view the historical details of all the previously generated test data for the Data definition. You can use this feature of Data fabrication to regenerate a test data for testing purposes.

See [Generating data from history of a Data definition on page 326](#).

Generating data from a Data definition

You can generate data from a Data definition using the Data fabrication feature of HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a Data definition. See [Creating an empty Data definition on page 251](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch from the **Branch** list.

The navigation panel displays all the existing projects in the selected branch.


5. Find the Data definition that you want to view, and then click it.



Note: You can either enter the name of the Data definition in the **Search** field or expand the project and manually search in the Data navigator panel.

Result


The selected Data definition opens in the **Data** page.

6. Click the **Preview** icon  to preview the test data that you want to generate.

The test data is generated in the preview dialog.




Note: You can preview a test data to ensure whether the data is in the expected format. You can have a glimpse of the generated test data that can help you to identify any potential issues, inconsistencies, or errors in the Data definitions. If you modify the data definition but not save it, you can still preview the corresponding test data. However, you can generate the test data only after you save the data definition.

7. Click the **Generate** icon  either from the **Data** page or from the preview page.

The **Generate data** dialog is displayed.



Note: If you are in an Edit branch, then you can access the **Generate** icon  from the preview dialog only if the data definition is saved.

8. Perform the following actions in the **Generate data** dialog:

- a. Enter a name in the **Name** field to identify the test data when generated.
- b. Enter the number of rows that you want to generate in the **Number of Rows** field.
- c. **Optional:** Enter the seed value in the **Seed** field.

Whenever you generate test data to perform application testing, the generated data is different. You can produce the same set of data multiple times by setting the seed value.

For example, assume that you have used the seed value as `1` to generate test data from a data definition. When you want to create another test data with the same set of values, then you can provide the seed value as `1` during the creation of the test data.

- d. Clear the **Include headers** option if you do not want to include headers in the generated test data.
- e. Click the **Download** option to initiate the generation of the test data.

Results

The generated test data is downloaded into your local file system as a Comma Separated Values (CSV) file.



Note: You can use other applications that support viewing or editing of the CSV files. You must ensure to apply the relevant formats to the fields to view the correct format of the generated test data.

What to do next

You can view the information about the history of the generated data in the **History** tab of the **Data** page.

Generating data from history of a Data definition

You can generate data from the history of a Data definition using the Data fabrication feature of HCL DevOps Test Hub (Test Hub) to regenerate test data by using the already created Data definitions. This feature also enhances collaboration and sharing when multiple project members work with a single test data at different times.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a Data definition. See [Creating an empty Data definition on page 251](#).
- Generated test data by using a Data definition. See [Generating data from a Data definition on page 324](#).



Note: If a project member has generated a test data from a Data definition, then the other project members can view the generated test data only when the changes are pushed to the repository.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Select the branch from the **Branch** list.

The navigation panel displays all the existing projects in the selected branch.

5. Find the Data definition that you want to use for regenerating test data, and then click it.



Note: You can either enter the name of the Data definition in the **Search** field or expand the project and manually search in the Data navigator panel.

Result

The selected Data definition opens in the **Data** page.


6. Click the **History** tab.

A list of all the previously generated test data are displayed, along with the following information:

- The month, date, and year of the generated test data.
- The status of the download action that indicates whether the data generation was successful or failed.
- The number of rows in the generated test data.
- The initials of the project member who downloaded the generated test data.

7. Perform the following actions in the **History** tab:

- a. Find the required test data.

- b. **Optional:** Click the **Preview** icon  to preview the test data before regeneration.

The test data is generated in the preview dialog.




Note: You can preview the previously generated test data to ensure whether the data is in the expected format. You can have a glimpse of the regenerated test data that can help you to identify any potential issues, inconsistencies, or errors in the Data definitions.

- c. Click the **Generate** icon .

The **Generate data** dialog is displayed.



Note: If you are in an Edit branch, then you can access the **Generate** icon  from the preview dialog only if the Data definition is saved.

8. Perform the following actions in the **Generate data** dialog:
 - a. Enter a name in the **Name** field to identify the test data when generated.
 - b. Enter the number of rows that you want to generate in the **Number of Rows** field.
 - c. **Optional:** Enter the seed value in the **Seed** field.

Whenever you generate test data to perform application testing, the generated data is different. You can produce the same set of data multiple times by setting the seed value.

For example, assume that you have used the seed value as 1 to generate test data from a data definition. When you want to create another test data with the same set of values, then you can provide the seed value as 1 during the creation of the test data.

- d. Clear the **Include headers** option if you do not want to include headers in the generated test data.
- e. Click the **Download** option to initiate the generation of the test data.

Results

The generated test data is downloaded into your local file system as a Comma Separated Values (CSV) file.



Note: You can use other applications that support viewing or editing of the CSV files. You must ensure to apply the relevant formats to the fields to view the correct format of the generated test data.

Management of custom generators

HCL DevOps Test Hub (Test Hub) provides flexibility to create, modify, and delete custom generators. You can access and modify custom generators to suit your specific data requirements in the **Catalog** page.

You can use the built-in Basic generators as the base to create custom generators that closely mimic the real-world data patterns and distributions. You can customize the parameters of the Base generators and save them as new custom generators to use them for future uses.

The creation, modification, and deletion of custom generators are restricted to the edit branch only. Also, only those members with either owner or tester roles within the project are authorized to create, modify, or delete these generators. This security measure can help to maintain data integrity and consistency while preventing unauthorized changes in Test Hub. After all the modifications are published, other project members can view the changes and updated assets.

Creating a custom generator

You can create custom generators in addition to the existing generators in the HCL DevOps Test Hub (Test Hub) catalog. You can customize the existing generators and save them as new custom generators to help you create complex real-world scenarios for synthetic data generation. Additionally, custom generators enable the generation of data for specialized fields or domains that may not be readily available in standard datasets.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Read information about the built-in basic generators. See [Basic generators on page 273](#).
- Read about the parameters of the basic generators. See [Basic generators on page 273](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Catalog**.

The **Catalog** page is displayed.

4. Select the branch where you want to store the data generator from the **Branch** list.

5. Perform one of the following actions to open the **Add new generator** dialog:

- Perform the following actions to use the base generator:
 - a. Click a generator from the **Basic** category that you want to use as the base generator.

The *Generator* dialog is displayed.

- b. Click the **Customize** icon .

- Click the  icon in the **Catalog** page, and then click **Add new generator**.

Based on the selected **Branch**, the following dialog is displayed:

- **Edit Branch:** If you selected a branch other than the Edit branch.
- **Add new generator:** If you selected the Edit branch.

6. Perform one of the following actions in the **Edit branch** dialog, if you selected a branch other than the Edit branch:

Action	Steps	Result
Select an Edit branch	<ol style="list-style-type: none"> a. Select an Edit branch from the Select branch list if the Edit branch already exists. b. Click OK. 	The Add new generator dialog is displayed.

Action	Steps	Result
Create an Edit branch	<ol style="list-style-type: none"> a. Select Create new branch. b. Enter a name for the Edit branch. c. Click Save. 	The Add new generator dialog is displayed.

7. Enter a unique name for the generator in the **Name** field of the **Add new generator** dialog.
8. Select a built-in generator from the **Base generator** list.
Test Hub displays only the built-in Basic generators as the base generator.
9. Perform one of the following actions to select the category of the new generator in the **Category** list:
 - To select an existing category, go to step [10 on page 330](#).
 - To add a new category, go to step [11 on page 330](#).
10. Perform the following steps to select an existing category:
 - a. Select an existing category, if any, that is displayed in the **Category** list.



Note: You cannot create a new generator in the **Basic** category. Therefore, the **Basic** category is not displayed in the **Category** list.

- b. Click **Create**.

The new generator is added under the existing category of the **Catalog** page.

11. Perform the following steps to add a new category:
 - a. Click **Add new category** in the **Category** list, and then click **Next**.

The **Add new category** dialog displays a new set of fields to capture more details about the new category.
 - b. Enter a unique name in the **Category name** field where the new generator will be saved in the catalog of Test Hub.
 - c. Select a color from the available range of colors so that you can identify the new category easily.
 - d. Select the location to create the new category in the selected folder.

For example, if you entered `MyFolder/MyCategory` in the **Category name** field, and then selected `Location` as the location to create the category in the folder, then the category is created under the `Location/MyFolder/MyCategory` path.
 - e. Click **Create**.

The new generator is added in the newly created category and is displayed in the **Catalog** page.

Results

You have created a custom generator in addition to the existing generators of the Test Hub catalog.

What to do next

You can perform any of the following tasks:

- Publish the custom generators to the project repository so that other members of the project can use the custom generator. See [Publishing changes in an Edit branch on page 267](#).
- Modify the parameters of custom generators. See [Modifying custom generators on page 332](#).
- Delete custom generators. See [Deleting custom generators on page 335](#).

Creating custom generators and categories from Data definitions

You can create custom generators and categories from existing Data definitions in HCL DevOps Test Hub (Test Hub). After you add an existing generator in a field of a Data definition, you can customize the generator parameters that are displayed for the field. You can then save the customized parameters in the Data fabrication catalog as a new generator. You can also save the new generator either in an existing category or save in a new category.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a Data definition. See [Creating an empty Data definition on page 251](#).
- Read about the parameters of the basic generators. See [Basic generators on page 273](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Data**.

The **Data** page and the Data navigator panel are displayed.

4. Open a Data definition in the Edit branch.
5. Click a generator in the **Generator** column of a field where you want to customize the configuration.
6. Perform any modification in the *Generator* dialog to customize the existing configuration of the selected field.
7. Click **Save to catalog** to save the customized parameters in the catalog of the Data fabrication feature as a new generator.

The **Add new generator** dialog is displayed.

8. Enter a unique name for the new generator in the **Name** field.

Test Hub uses the existing generator as the base generator. Therefore, the **Base generator** list is not editable.

9. Perform one of the following actions to select the category of the new generator:

- To select an existing category, go to step [10 on page 331](#).
- To add a new category, go to step [11 on page 332](#).

10. Perform the following steps to select an existing category:

- a. Select an existing category that is displayed in the **Category** list.
- b. Click **Create**.

The new generator is added in the existing category and is displayed in the **Catalog** panel of the data definition.

11. Perform the following steps to add a new category:

- a. Click **Add new category** in the **Category** list, and then click **Next**.

The **Add new generator** dialog displays a new set of fields to capture more details about the new category.

- b. Enter a unique name in the **Category name** field where the new generator will be saved in the catalog of Test Hub.
- c. Select a color from the available range of colors so that you can identify the new category easily.
- d. Select the location to create the new category in the selected folder.

For example, if you entered `MyFolder/MyCategory` in the **Category name** field, and then selected `Location` as the location to create the category in the folder, then the category is created under the `Location/MyFolder/MyCategory` path.

- e. Click **Create**.

The new generator is added in the newly created category and is displayed in the **Catalog** panel of the data definition.

Results

You have created a custom generator using an existing generator from a Data definition. Also, you have learned to create a new category and save the custom generator under the category.

What to do next

You can save and publish the Data definition to the Git repository so that other members of the project can use the Data definition. See [Publishing changes in an Edit branch on page 267](#).

Modifying custom generators

You can modify the parameters of custom generators from the Catalog page of HCL DevOps Test Hub (Test Hub). You can make modifications to precisely align the test data that uses the custom generators with the desired dataset.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a custom generator. See [Creating a custom generator on page 329](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Catalog**.

The **Catalog** page is displayed.


4. Select the branch that contains the custom generator for modification from the **Branch** list.

5. Select the custom generator for modification.

The selected *Generator* dialog that contains the following tabs is displayed:

Element	Description
The Configuration tab	Displays all the existing parameters of the selected custom generator.
The Details tab	Provides following details about the selected custom generator and its associated Git details: <ul style="list-style-type: none"> ◦ Name of the branch that contains the generator ◦ Description for the generator, if any ◦ Git-related details for the generator as follows: <ul style="list-style-type: none"> ▪ Asset ID: The ID uniquely identifies the generator. You also have the option to copy the ID for easy reference or use. ▪ Repo: The repository name and path where the generator is stored. ▪ Path: The exact location of the generator within the repository. ▪ Commit: The Git commit message, the name of the member who performed the changes, and the date and time of the commit. ▪ The copy button: You can click the button to copy the complete SHA commit ID for the generator. ▪ The commit ID button: You can click the button to view the changes that were made in the commit directly within Git. ▪ The browse button: You can click the button to view the specific location in the Git history where the commit was performed for the generator.
The Usage tab	Displays a list of all the Data definitions along with their location in the project that currently use the selected generator.


6. Perform one of the following actions in the **Generator** dialog:

- Perform the required modifications to the existing configurations in the **Configuration** tab, if you are in the edit branch.
- Perform the following actions, if you are in a branch other than the Edit branch:
 - a. Click the **Edit** icon .

The **Edit branch** dialog is displayed.

- b. Perform one of the following actions in the dialog:

Action	Steps	Result
Select an Edit branch	<ol style="list-style-type: none"> i. Select an Edit branch from the Select branch list if the Edit branch already exists. ii. Click Ok. 	The Catalog page is displayed in the Edit branch. You can open the custom generator and perform modifications to the existing generator configurations in the Configuration tab of the <i>Generator</i> dialog.
Create an Edit branch	<ol style="list-style-type: none"> i. Select Create new branch. ii. Enter a name for the Edit branch. iii. Click Save. 	The Catalog page is displayed in the Edit branch. You can open the custom generator and perform modifications to the existing generator configurations in the Configuration tab of the <i>Generator</i> dialog.

7. Click the save icon  in the *Generator* dialog to save the modified custom generator.



Note: If you want to continue using the modified generator to generate test data without publishing it, you must save the changes. However, if you want to collaborate with other members of your project, you must publish the changes.

Results

You have modified the parameters of a custom generator.

What to do next

You can perform any of the following tasks:

- Publish the custom generators to the project repository so that other members of the project can use the custom generator. See [Publishing changes in an Edit branch on page 267](#).
- Delete custom generators. See [Deleting custom generators on page 335](#).

Deleting custom generators

You can delete custom generators to remove them when custom generators are outdated or no longer relevant to the current project. Additionally, when custom generators contain errors or inaccuracies that can lead to misleading or incorrect fabricated data, then you might need to remove them. This ensures that the project remains clutter-free, optimizes the performance of HCL DevOps Test Hub (Test Hub), and reduces storage requirements.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a custom generator. See [Creating a custom generator on page 329](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Catalog**.

The **Catalog** page is displayed.

4. Select the branch that contains the custom generator to delete from the **Branch** list.

5. Select the custom generator for deletion.

The selected *Generator* dialog is displayed.

6. Click the **Delete** icon .

Results

You deleted a custom generator from your project.

What to do next

You can publish the changes performed to the custom generator (that is, deletion of the custom generator) to the Git repository so that other members of the project do not see the deleted custom generator. See [Publishing changes in an Edit branch on page 267](#).

Management of custom categories

HCL DevOps Test Hub (Test Hub) provides flexibility to create and delete custom categories in the **Catalog** page. The **Catalog** page displays both the basic and custom categories of generators in a project.

The basic category consists of all the built-in basic generators to simplify the access when you work with Data definitions. When you create custom generators that are tailored to specific use cases, then you can save them under the custom categories and organize them in a way that aligns with the project requirements. You can locate and use the custom generators when you need.

The creation and deletion of custom categories are restricted to the Edit branch only. Also, only those members with either owner or tester roles within the project are authorized to create or delete these categories. This security measure can help to maintain data integrity and consistency while preventing unauthorized changes in Test Hub. After all the modifications are published, other project members can view the changes and updated assets.

Creating a new custom category

You can create custom categories in addition to the existing Basic category in the HCL DevOps Test Hub (Test Hub) catalog. You can save the custom generators under various custom categories with unique names.

Before you begin

You must be a member of the project with the **Owner** or **Tester** role.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Catalog**.

The **Catalog** page is displayed.

4. Select the branch where you want to store the custom category from the **Branch** list.

5. Click the  icon in the **Catalog** page, and then click **Add new category**.

Based on the selected **Branch**, the following dialog is displayed:

- **Edit Branch:** If you selected a branch other than the Edit branch.
- **Add new category:** If you selected the Edit branch.

6. Perform one of the following actions in the **Edit branch** dialog, if you selected a branch other than the Edit branch:

Action	Steps	Result
Select an Edit branch	<ol style="list-style-type: none"> Select an Edit branch from the Select branch list if the Edit branch already exists. Click Ok. 	The Add new category dialog is displayed.
Create an Edit branch	<ol style="list-style-type: none"> Select Create new branch. Enter a name for the Edit branch. Click Save. 	The Add new category dialog is displayed.

7. Perform the following steps in the **Add new category** dialog:

- Enter a unique name in the **Category name** field.
- Select a color from the available range of colors so that you can identify the new category easily.
- Select the location to create the new category in the selected folder.

For example, if you entered `MyFolder/MyCategory` in the **Category name** field, and then selected `Location` as the location to create the category in the folder, then the category is created under the `Location/MyFolder/MyCategory` path.

d. Click **Create**.

The new category is added in the **Catalog** page.

Results

You created a custom category in the **Catalog** page.

What to do next

You can perform the following tasks:

- Add custom generators to the new category. See [Creating a custom generator on page 329](#).
- Publish the changes to the Git repository so that other members of the project can use the new custom category. See [Publishing changes in an Edit branch on page 267](#).

Deleting custom categories

You can delete custom categories to remove them when custom categories are outdated or no longer relevant to the current project. Additionally, when custom categories contain inaccuracies that can lead to misleading or incorrect fabricated data, then you might need to remove them. This ensures that the project remains clutter-free, optimizes the performance of HCL DevOps Test Hub (Test Hub), and reduces storage requirements.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a custom category. See [Creating a new custom category on page 336](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Author > Catalog**.

The **Catalog** page is displayed.

4. Select the branch that contains the custom category that you want to delete from the **Branch** list.

5. Click the **Delete** icon  beside the category tab to delete the custom category.

Based on the selected **Branch**, the following dialog is displayed:

- **Edit Branch:** If you selected a branch other than the Edit branch.
 - **Delete category:** If you selected the Edit branch.
6. Perform one of the following actions in the **Edit branch** dialog, if you selected a branch other than the Edit branch:

Action	Steps	Result
Select an Edit branch	<ul style="list-style-type: none"> ◦ Select an Edit branch from the Select branch list if the Edit branch already exists. ◦ Click Ok. 	The Delete category dialog is displayed.
Create an Edit branch	<ul style="list-style-type: none"> ◦ Select Create new branch. ◦ Enter a name for the Edit branch. ◦ Click Save. 	The Delete category dialog is displayed.

7. Click **Delete** in the **Delete category** dialog.

Results

You deleted a custom category from your project.

What to do next

You can publish the changes performed to the custom category (that is, deletion of the custom category) to the Git repository so that other members of the project do not see the deleted category. See [Publishing changes in an Edit branch on page 267](#).

Chapter 6. Test Execution Specialist Guide

This guide describes tasks that you must complete before you can configure and run tests in HCL DevOps Test Hub (Test Hub). You can find information about configuring runs for the different test types that are supported. You can also find information about other tasks that you can perform on the Resource Monitoring Service, Virtualization, and Integrations with third-party applications. This guide is intended for testers or test execution specialists.

System modeling

You can use the system model feature on HCL DevOps Test Hub (Test Hub) to create a model that simulates an application or system under test. You can define components to represent the actual components in the system under test, define their relationships and then view the components in a system under test as a diagrammatic representation.

Introduction to a system model

A system model is a logical presentation of the components contained in a system under test and the relationships between components. You can create a system model when you want to visualize the different components and their relationships as a diagrammatic representation.

You can create only one system model within a team space on Test Hub. A team space repository is required to contain the system model that you create. Before you can create a system model, you must add a team space repository. The system model that is created or edited is stored in the team space repository. All projects within a team space share the same system model. All members of a team space can create, edit, and view the system model.

After you log in to Test Hub, you can go to the **System Model** page to create a system model for your team space. Alternatively, you can go to any of the projects in your team space and create a system model from the **System Model** page within a project. The system model is accessible from all the projects in your team space.

Components in a system model

Components are the basic building blocks that are used to represent a piece of software in an application.

You can model a system by using components. You can create components that represent the type of asset or resource used by the system or application under test and can be of the following types:

Component type	Description
Database	The component type represents the database resource. You can select this component type when you want to represent a database asset or resource in the system model.
Service	The component type represents the virtual service resource. You can select this component type in the system model to represent a virtual service in the team space repository or associate a virtual service resource that is in the project repository.

Component type	Description
UI	The component type represents the user interface (UI) resource. You can select this component type when you want to represent the UI resource in the system model.

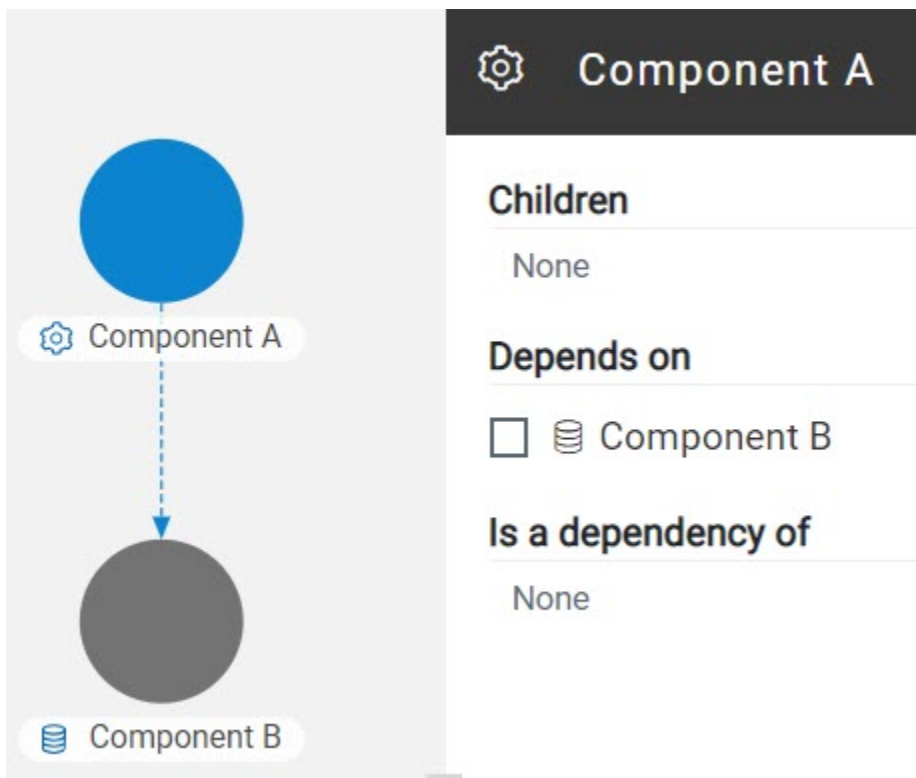
Component relationships

You can create multiple components to depict the different resources in the system or application under test. You can create a component as a *child* component of a *parent* component. You can also create multiple components as children components of the same parent component.

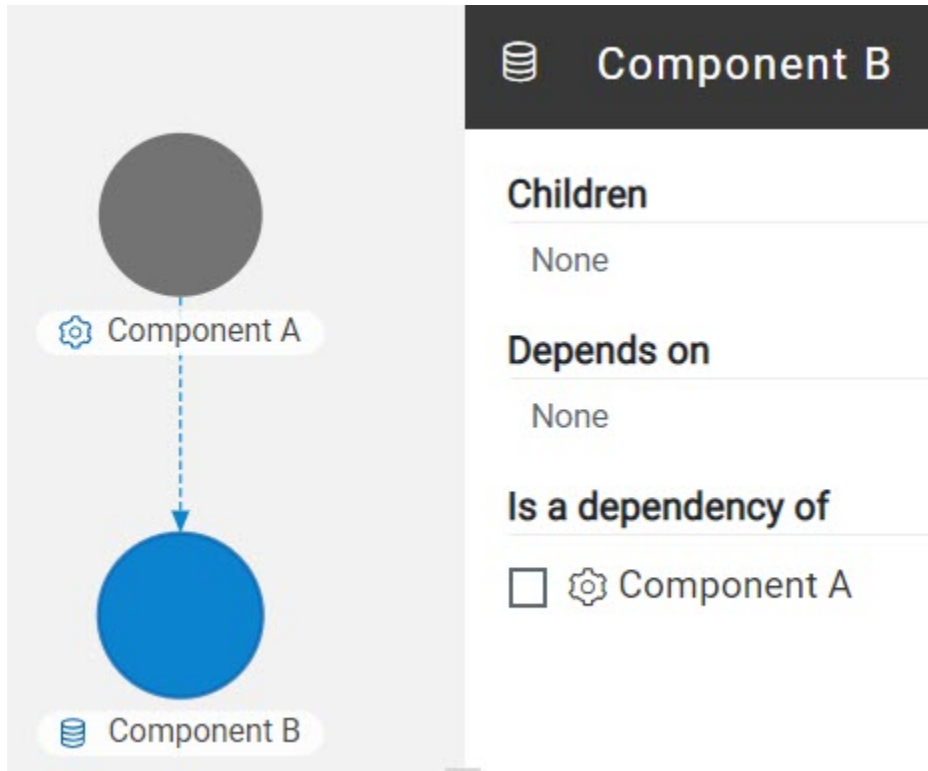
You can define the relationship between components if one component depends on another or if a component has a dependency of another component. The **Depends on** or **Is a dependency of** relationship depends on the component that you select in the system model after you define the relation.

For example, if you define *Component A* to depends on *Component B*, then the relationship is displayed with a line arrow from *Component A* to *Component B*, and the arrow points to *Component B*.

When you click the *Component A* in the system model page, you can view that the *Component B* is listed under the **Depends on** section for *Component A* in the right pane.



When you click the *Component B* in the system model, you can view that the *Component A* is listed under the **Is a dependency of** section for *Component B* in the right pane.



Component associations






When you work with the system model from your project in a team space, you can associate test resources that are in your project repository with components in the system model. You can associate all types of test resources such as Suites or tests, and virtual service resources with the components.

After associating test resources, you can view them when you select the component with which they are associated. You can select the associated test resources to start a run of the Suites or tests or start an instance of the virtual services resource.




Using the system model UI

The following table lists the icons used in a system model and the icons that are displayed on the **System Model** page:



Icon	Description
	Represents a system model.
	Represents Suites and tests in a repository.
	Represents a database resource. The component name is prefixed with this icon when you select the component type as database.





Icon	Description
	Represents a service resource. The component name is prefixed with this icon when you select the component type as service.
	Represents a user interface (UI) resource. The component name is prefixed with this icon when you select the component type as UI.
	Represents the virtual services.
	Represents the dependency between the components.
	Represents the inherited dependencies.

The following table lists the icon that you can use to perform different actions in the system model:

Icon	Used for...
	Adding components, child components, or dependent components.
	Searching for components in the system model.
	Publishing the changes made to the system model, changes made to components, or changes made to component associations. The changes are published to the team space repository.

You can perform the following operations when you view the model that is displayed on the **System Model** page:

Operation	Action
Pan the view of the model	<ol style="list-style-type: none"> 1. Click the area of the model that you want to pan and move the area by keeping the left mouse button pressed. 2. Release the left mouse button to stay on the view that you panned.
Zoom in the view of the model	<ol style="list-style-type: none"> 1. Click the Zoom-in icon . <p>The view is enlarged or zoomed in.</p> <p> Note: You can repeatedly click the Zoom-in icon to further enlarge the view.</p> <p>Alternatively, you can use the scroll button on the mouse or device to zoom in the view.</p>

Operation	Action
Zoom out the view of the model	<p>1. Click the Zoom-out icon .</p> <p>The view is diminished or zoomed out.</p> <p> Note: You can repeatedly click the Zoom-out icon to further diminish the view.</p> <p>Alternatively, you can use the scroll button on the mouse or device to zoom out the view.</p>
Reset the view	<p>1. Click the Menu icon  on the System Model page.</p> <p>2. Select Reset Zoom.</p> <p>The view is reset to the default view without any zoom applied. If the view was zoomed in or zoomed out, the view is reset to 100% of the size.</p>
Position the model view	<p>1. Click the Menu icon  on the System Model page.</p> <p>2. Select from the following options:</p> <ul style="list-style-type: none"> ◦ Center: Select this option to align the view of the model to the center of the pane. ◦ Fit to View: Select this option to fit the view of the model in the pane on the System Model page.

Tasks in system modeling

When you want to model the system under test for the first time after you install the server software, you must first add the team space repository. See [Adding a repository to a team space on page 345](#).

After a repository is configured as the team space repository, you can work with the system model. See [Tasks for working with a system model on page 344](#).

When you want to change, update or delete the repository, you can work with the team space repository. See [Tasks for working with the team space repository on page 345](#).

Task flows for working with a system model

You can find information about the tasks that you can perform when you want to create a system model, components, associate the components, create linkages among the components, and then publish the system model

to the team space repository. You can also find information about the tasks that you must perform when you want to modify an existing system model.

Tasks for working with a system model

You can find the tasks when you want to create a system model for the application under test in your team space.

Tasks	Go to...
Add a repository to the team space.	Adding a repository to a team space on page 345
Create a system model.	Creating a system model on page 347
Create components in a system model.	Creating components on page 349
Create child components to existing components in a system model.	Creating child components on page 350
Define the relationship between the components in a system model.	Creating linkages between components on page 352
Delete the relationship between the components in a system model.	Deleting linkages between components on page 354
Modify components in a system model.	Modifying components on page 356
Delete components in a system model.	Deleting components on page 358
Publish changes made to the system model.	Publishing changes to the system model on page 359
Associate resources with components in a system model.	Associating resources with components on page 359
View resources associated with components in a system model.	Viewing resources that are associated with components on page 361
Remove resources that are associated with components in a system model.	Removing resources that are associated with components on page 365
Publish changes of associating or dissociating resources in a project with components in the system model.	Publishing changes of associating or dissociating resources with components on page 367
View system model.	Viewing the system model on page 369

Tasks	Go to...
Delete system model.	Deleting a system model on page 369

Tasks for working with the team space repository

You can find the tasks that you can perform when you want to work with the team space repository.

Tasks	More information
Add a repository to the team space.	Adding a repository to a team space on page 345
Delete a team space repository.	Deleting a repository that is added to a team space on page 370

Adding a repository to a team space

You must add a repository to the team space in HCL DevOps Test Hub (Test Hub) when you want to create, modify, or store a system model.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* or an *Architect* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you obtained valid credentials to access the Git repository that you want to add.

About this task


As a *Team Space Owner* or an *Architect*, you can add a repository to the team space. You can add a repository to the team space in any of the following methods:

- Add an existing repository that can contain test assets, which are created in the desktop products.
- Add a bare repository.




Attention: The repository that you want to add must not contain any projects or `.project` files under the root directory.

When you add a repository, the Git repository is cloned to the team space. While you add a repository, you must provide the necessary authentication credentials that are set for the Git repository. For example, if the authentication type is SSH, then you must provide the Git URL, a deploy key, and a passphrase.

 **Important:** You can add only one repository to the team space. After you add a repository to the team space, the option to add another repository is not available. You must delete an existing repository before you can add a different repository.

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which you want a team space repository.

The **Active projects** page is displayed.

3. Click the **Settings** icon  from the left navigation pane, and then select **Manage team space**.

The **Details** page is displayed.

4. Click **Repository** on the left navigation pane.

The **Repository** page is displayed.

5. Click **Add repository**.

The **Add repository** dialog is displayed.

6. Enter the URL of the Git repository, which you want to add to the team space, in the **Git Repository** field.

The required fields are displayed based on the type of Git URL that you entered.

7. Enter the required credentials based on any of the following authentication methods configured in the repository.

To gain access to the repository, you must use any one of the authentication methods:

Authenticati- on method	Credentials required
SSH	<ul style="list-style-type: none"> ◦ Deploy key ◦ Passphrase
HTTPS	<ul style="list-style-type: none"> ◦ Username ◦ Password
HTTP	<ul style="list-style-type: none"> ◦ Username ◦ Password

 **Notes:**



- You must have defined the authentication type and set the authentication credentials in the Git repository.
- If you use SSH to connect to your remote repository and Test Hub displays an `Auth Fail` exception while using the `deliver changes` option, you can resolve this exception error by regenerating your SSH keys by using the `-m PEM` option.

8. Click **Add**.

The Git repository is added to the team space on Test Hub.



Note: The time to clone the repository can take a few to several minutes and depends on the size of the repository.

What to do next

You can perform the following actions on the repository that you added:

- Update the authentication credentials if they are changed in the Git repository configuration. See [Updating the authentication credentials of the repository on page 1030](#).
- Refresh a repository to fetch and synchronize changes from the remote repository. See [Refreshing repositories manually on page 1031](#).
- Configure a webhook to notify the server if there is a push event in the remote repository. See [Creating webhooks on page 1034](#).
- Add a system model to the repository. See [Creating a system model on page 347](#).
- Delete a repository that is added to the team space, if it is no longer required. See [Deleting a repository that is added to a team space on page 370](#).

Related information

[Managing repositories on page 1026](#)

Creating a system model

Before you can create components in a system model in HCL DevOps Test Hub (Test Hub), you must create an empty system model and publish the system model to the team space repository.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as an *Architect* in the team space, a *Project Owner* or a *Tester* in the project. See [Managing members and their roles in a team space on page 1006](#) or [Managing access to the server projects on page 1021](#).
- Added a repository to the team space. See [Adding a repository to a team space on page 345](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Select an option to create a system model based on your role in the team space or in a project in the team space:

- If you are a *Team space owner* or an *Architect*, go to [Step 4 on page 348](#).
- If you are not a *Team space owner* or an *Architect* but a project *Owner* or *Tester*, go to [Step 5 on page 348](#).

4. Perform the following steps, if you are a *Team space owner* or an *Architect*:

- a. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.

- b. Click the **Add system model** icon .

An empty system model is created.

- c. Go to [Step 6 on page 348](#).

5. Perform the following steps, if you are not a *Team space owner* or an *Architect* but a *Project owner*:

- a. Open your project that is displayed under **My projects** on the **Active projects** page.

The **Overview** page is displayed.


- b. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.

- c. Click the **Add system model** icon .

An empty system model is created.

- d. Go to [Step 6 on page 348](#).

6. Click the  icon, and then click the **Publish changes** option.



Note: If you do not want to commit the changes that you made to the system model, you can select the **Discard changes** option.

7. Enter a commit message in the **Description of change** field, and then click **Publish**.

The empty system model is committed and published to the team space repository. When the system model is published to the team space repository, other members of the team space can work with the same system model from any of their projects in the team space.

Results

You have created an empty system model and published it to the team space repository.

What to do next

You can perform any of the following tasks:

- Create components in the system model. See [Creating components on page 349](#).
- Create child components for the components that you added to the system model. See [Creating child components on page 350](#).

Creating components

After you create an empty system model in the team space in HCL DevOps Test Hub (Test Hub), you can create components that represent components of the system under test.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as an *Architect* in the team space, a *Project Owner* or a *Tester* in the project. See [Managing members and their roles in a team space on page 1006](#) or [Managing access to the server projects on page 1021](#).
- Added a repository to the team space. See [Adding a repository to a team space on page 345](#).
- Created and published the system model. See [Creating a system model on page 347](#).


1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.




4. Perform the following steps to add a component to the system model:

- a. Click the **Add component** icon  in the **Components** row on the right pane.

The **Add a component** dialog is displayed.

- b. Enter a name for the component in the **Name** field.

- c. Select the type of component that you want to add from the following options displayed in the **Type** field.

-  **Service**
-  **Database**
-  **UI**

d. Click **Add**.

The component is displayed as a circular block on the system model pane with the icon and name of the component below the block. The added component is also displayed under the **Components** list on the right pane.

You can create components of other types or the same type in the system model by following the preceding steps.

Results

You have created components in the system model.

What to do next

You can perform any of the following tasks:

- Publish the changes that you made to the system model. See [Publishing changes to the system model on page 359](#).
- Create child components for the components that you added to the system model. See [Creating child components on page 350](#).
- Create linkages between components in the system model. See [Creating linkages between components on page 352](#).

Creating child components

You can create child components in the system model in HCL DevOps Test Hub (Test Hub) when you want to represent the sub-components of a component in a system under test.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as an *Architect* in the team space, a *Project Owner* or a *Tester* in the project. See [Managing members and their roles in a team space on page 1006](#) or [Managing access to the server projects on page 1021](#).
- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).

About this task

You can create multiple components in your system model and multiple child components under any component in the system model. After you create a child for a component, the child is represented as a circular block within the circular component block in the system model.

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.


The **Active projects** page is displayed.

3. Click **System Model** on the left navigation pane.

The **System Model** page is displayed with the existing components.




4. Create child components by using any of the following options:
 - To create child components from the left pane on the **System Model** page, go to [Step 5 on page 351](#).
 - To create child components from the right pane on the **System Model** page, go to [Step 6 on page 351](#).
5. Perform the following steps to create child components from the left pane:
 - a. Click the component under which you want to create child components.

The details of the component are displayed in the right pane.

- b. Click the **Add child** icon  in the **Children** row on the right pane.

The **Add a child to...** dialog is displayed.

- c. Enter a name for the component in the **Name** field.
 - d. Select the type of component that you want to add from the following options displayed in the **Type** field.


-  **Service**
-  **Database**
-  **UI**




- e. Click **Add**.

The child component is displayed as a circular block within the circular block of the parent component on the system model pane. The added child component is also displayed under the **Children** list under the component on the right pane.

You can create child components of other types or the same type for the same or other components in the system model by following the preceding steps.

6. Perform the following steps to create child components from the right pane of the stem model page:

- a. Click the component name listed under the **Components** in the right pane.
- b. Click the **Add child** icon  in the **Children** row on the right pane.
The **Add a child to...** dialog is displayed.
- c. Enter a name for the component in the **Name** field.
- d. Select the type of component that you want to add from the following options displayed in the **Type** field.

-  **Service**
-  **Database**
-  **UI**

- e. Click **Add**.

The child component is displayed as a circular block within the circular block of the parent component on the system model pane. The added child component is also displayed under the **Children** list under the component on the right pane.

You can create child components of other types or the same type for the same or other components in the system model by following the preceding steps.

Results

A component is created and it is associated as a child with the selected component and the name of the child is displayed under the **Children** section on the right pane.

What to do next

You can perform any of the following tasks:

- Publish the changes that you made to the system model. See [Publishing changes to the system model on page 359](#).
- Create linkages between components in the system model. See [Creating linkages between components on page 352](#).

Creating linkages between components

You can create linkages between components in HCL DevOps Test Hub (Test Hub) to display the relationship between them. The linkages can be between components or between child components of the same or different parent components.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as an *Architect* in the team space, a *Project Owner* or a *Tester* in the project. See [Managing members and their roles in a team space on page 1006](#) or [Managing access to the server projects on page 1021](#).
- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).
- Created child components. See [Creating child components on page 350](#).

About this task

You can create multiple components in your system model. You can also create multiple child components under any component in the system model. You can specify the dependency among the components.

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **System Model** on the left navigation pane.


The **System Model** page is displayed.


4. Perform the following steps to create a relationship between the components in the system model:

- a. Click the component that you want.

The details of the component are displayed on the right pane.

- b. Perform the actions in the following table to create dependencies between the components:

Option	Action
Depends on	<p>Perform the following steps, if the component that you selected is dependent on another component:</p> <ol style="list-style-type: none"> i. Click the Add dependency icon  in the Depends on row on the right pane. <p>The Add dependency dialog is displayed.</p> <ol style="list-style-type: none"> ii. Select the component from the list of components displayed under Depends on. iii. Click Add. <p>The components linked are listed under the Depends on section on the right pane. The linkage between the first to the second component that you selected is shown with an arrow that starts from the first and ends in the second component.</p>

Option	Action
Is a dependency of	<p>Perform the following steps, if the component that you selected has a dependency on another component:</p> <ol style="list-style-type: none"> i. Click the Add dependent icon  in the Is a dependency of row on the right pane. <p>The Add dependent dialog is displayed.</p> <ol style="list-style-type: none"> ii. Select the component from the list of components displayed under the Is a dependency of section. iii. Click Add. <p>The components linked are listed under Is a dependency of section on the right pane. The linkage between the first to the second component that you selected is shown with an arrow that starts from the second and ends in the first component.</p>

Results

You have created linkages between components in the system model.

What to do next

You can publish the changes that you made to the system model. See [Publishing changes to the system model on page 359](#).

Deleting linkages between components

When you do not want to retain the linkages between components that you created in the system model in HCL DevOps Test Hub (Test Hub), you can delete the linkages.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as an *Architect* in the team space, a *Project Owner* or a *Tester* in the project. See [Managing members and their roles in a team space on page 1006](#) or [Managing access to the server projects on page 1021](#).
- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).
- Created child components. See [Creating child components on page 350](#).
- Created linkages between components in the system model. See [Creating linkages between components on page 352](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **System Model** on the left navigation pane.



The **System Model** page is displayed.

4. Perform the following steps to delete a relationship between the components in the system model:

- a. Click the component that you want.

The details of the component are displayed on the right pane.

- b. Select from the following types of relationships that you want to delete:

When...	Action
The component depends on another component.	<p>Perform the following steps, if the component that you selected is dependent on another component:</p> <ol style="list-style-type: none"> i. Select the component from the list of components that are displayed under the Depends on section on the right pane. ii. Click the Delete selected icon  inline with the Depends on row. <p>The Delete relationship dialog is displayed with the details of the component and the dependent component.</p> <ol style="list-style-type: none"> iii. Click Delete. <p>The component linkage is removed from the system model pane.</p>
The component has a dependency on another component.	<p>Perform the following steps, if the component that you selected has a dependency on another component:</p> <ol style="list-style-type: none"> i. Select the component from the list of components that are displayed under the Is a dependency of section on the right pane. ii. Click the Delete selected icon  inline with the Is a dependency of row. <p>The Delete relationship dialog is displayed with the details of the component and the component dependency.</p> <ol style="list-style-type: none"> iii. Click Delete. <p>The component linkage is removed from the system model pane.</p>

Results

You have created linkages between components in the system model.

What to do next

You can publish the changes that you made to the system model. See [Publishing changes to the system model on page 359](#).

Modifying components

At any point in time, after you create a component in HCL DevOps Test Hub (Test Hub), you can modify and update the details of the components such as the component type, and the relationships with other components.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as an *Architect* in the team space, a *Project Owner* or a *Tester* in the project. See [Managing members and their roles in a team space on page 1006](#) or [Managing access to the server projects on page 1021](#).
- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).
- Created child components. See [Creating child components on page 350](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.



3. Click **System Model** on the left navigation pane.





The **System Model** page is displayed.

4. Click the component that you want to edit.

The component details are displayed on the right pane.

5. Perform the actions for the task to modify components as listed in the following table:

Task	Action
Change the component type.	<p>Perform the following steps:</p> <ol style="list-style-type: none"> a. Identify the component that you want to modify from the components listed under Components on the right pane. b. Click the component checkbox. <p>The Edit selected icon  is enabled.</p> <ol style="list-style-type: none"> c. Click the Edit icon .

Task	Action
	<p>The Edit component dialog box is displayed.</p> <p>d. Select the component type.</p> <p>e. Click Modify</p> <p>The type of the component is modified.</p>
Change the name of a component.	<p>Perform the following steps:</p> <p>a. Identify the component that you want to modify from the components listed under Components on the right pane.</p> <p>b. Click the component checkbox.</p> <p>The Edit selected icon  is enabled.</p> <p>c. Click the Edit icon .</p> <p>The Edit component dialog box is displayed.</p> <p>d. Edit the component name in the Name field.</p> <p>e. Click Modify.</p> <p>The name of the component is modified.</p>
Change the parent of a component.	<p>Perform the following steps:</p> <p>a. Identify the component that you want to modify from the components listed under Components on the right pane.</p> <p>b. Click the component checkbox.</p> <p>The Change parent icon  is enabled.</p> <p>c. Click the Change parent icon .</p> <p>The Change parent dialog box is displayed.</p> <p>d. Select another parent for the selected components from those that are listed under the Parent list.</p> <p>e. Click Change.</p> <p>The parent of the component is changed.</p>

Results

You have edited components and updated the details successfully.

What to do next

You can publish the changes that you made to the system model. See [Publishing changes to the system model on page 359](#).

Deleting components

When you do not require a component in HCL DevOps Test Hub (Test Hub), you can delete the component from the system model.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as an *Architect* in the team space, a *Project Owner* or a *Tester* in the project. See [Managing members and their roles in a team space on page 1006](#) or [Managing access to the server projects on page 1021](#).
- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.

4. Click the component that you want to delete.

The component details are displayed on the right pane.

5. Click the **Delete component** icon  in the row of the component name.

The **Delete component** dialog is displayed.

6. Click **Delete**.

Results

You have deleted a component from the system model.

What to do next

You can perform any of the following tasks:

- Publish the changes that you made to the system model. See [Publishing changes to the system model on page 359](#).
- Create components in the system model. See [Creating components on page 349](#).

Publishing changes to the system model

When you create, delete, or modify components in a system model in HCL DevOps Test Hub (Test Hub), the changes are saved locally and are not visible to the other project members or team space members. You must publish the system model changes to the team space repository to make your changes visible to all users of the team space.

Before you begin

You must have completed the following tasks:



- Ensured that you are assigned a role as an *Architect* in the team space, a *Project Owner* or a *Tester* in the project. See [Managing members and their roles in a team space on page 1006](#) or [Managing access to the server projects on page 1021](#).
- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).
- Created child components. See [Creating child components on page 350](#).
- Created linkages between components in the system model. See [Creating linkages between components on page 352](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.

4. View the number of unpublished changes made to the system model that is displayed along with the  icon.
5. Click the  icon, and then click the **Publish changes** option.



Note: If you do not want to commit the changes that you made to the system model, you can select the **Discard changes** option.

6. Enter a commit message in the **Description of change** field, and then click **Publish**.

Results

You have published the changes that you made in the system model to the team space repository. Other members of the team space can view and use the system model.

Associating resources with components

After creating components in the system model in HCL DevOps Test Hub (Test Hub), you can navigate to a project in the team space to associate the Suites, tests, or virtual service resources in the project repositories with components in the system model.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

4. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.

5. Click the component that you want to associate with the Suites, Tests, or virtual service resources.

The component details are displayed in the right pane.

6. Click the **Associate** tab, and then perform the following steps:

- a. Click **Add**.

- b. Select the type of resource that you want to associate with the component from the following options:

- Suites and Tests
- Virtual services



Note: You can select either of the options or both the options.

The **Edit branch** dialog is displayed.

- c. Select your action that depends on the existence of an *Edit* branch from the following options:

- Select an existing *Edit* branch that you created previously from the list of branches listed in the **Select branch** field, and then click **Save**.
- Click **Create new branch**, enter a name for the branch, and then click **Save**, if you want to create a new branch.

The **Associate Assets** dialog is displayed. The resources that are contained in the branch of the project repository are listed. The name of the component and the branch are also displayed.

The **Suites and Tests** option is selected and the assets are listed if you selected the **Suites and Tests** option.


The **Virtual services** option is selected and the virtual service resources are listed if you selected the **Virtual services** option.

If you selected both the types of resources, then assets and virtual services in the branch are listed.

- d. Identify the assets or virtual resources that you want to associate with the component in the system model by performing any of the following actions:
 - Search for the resource by entering the asset name in the **Search** field.
 - Find the assets by scrolling through the list of assets.
- e. Select the resources, and then click **Add**.

The selected resources are associated with the component.

You can view the associated resources that are displayed in the **Associate** tab and the **Overview** tab. The test assets are listed under the **Suites and Tests** section and the virtual service resources are listed under the **Virtual services** section.

If you want to associate other resources to the same component, you can add them by clicking the **Add association** icon  in the resource row on the right pane. You can then select the resource and add the resource to the selected component.

Results

You have associated Suites, tests, or virtual service resources with a component in the system model.

What to do next

You can perform any of the following tasks:

- Publish the changes made in the system model for associating resources with components. See [Publishing changes to the system model on page 359](#).

Viewing resources that are associated with components

After you associate Suites, tests, or virtual services resources with components in the system model in HCL DevOps Test Hub (Test Hub), you can view the associated resources from the **System Model** page. You can opt to run an instance of the associated resource or stop a running instance of the associated resource.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).
- Associated resources in a project with components in the system model. See [Associating resources with components on page 359](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

4. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.




5. Click the component that you associated with the resources in the project repository.





The resources associated with the component are displayed under the resource type in the **Associate** tab.

6. Click the **Overview** tab, if it is not already displayed.






The resources that are associated with the component are displayed under the **Suites and Tests** or **Virtual Services** section.





7. View and perform the actions for the resource type that you want from the following options:
 - For **Suites and Tests** resources, go to [8 on page 362](#).
 - For **Virtual Services** resources, go to [9 on page 363](#).
8. Identify the test resource from those that are listed under the **Suites and Tests** section, and perform the action that you want as described in the following table:



Task	Description	Action
Collapse icon 	You can hide the resource sections by clicking the Collapse icon  .  Note: When the section is hidden, you can view the number that is dis-	Click the icon to toggle the display to show or hide the resource list in the Suites and Tests section.

Task	Description	Action
	 played of the resources that are associated.	
Type	You can view the type of Suite or test.	Hover over the Suites and Tests icon under the Type column to view the type of Suite or test.
Name	You can view the name of the associated Suite or test and the path in the repository.	There is no action to perform.
Status	You can view the details of the test runs.	Click each of the squares to view the details of the specific test run. For example, if the status of a test run is represented as shown in the image  , then it indicates that the Suite or test has run only two times. After you click any colored square, the Results page is displayed.
Action	You can start the execution of the Suite or test from the Execution page when you click the Show in execution page icon  .	Click the Show in execution page icon  to view the Suite or test on the Execution page. The Suite or test is displayed on the Execution page.

9. Identify the virtual service resource from those that are listed under the **Virtual Services** section, and perform the action that you want as described in the following table:

Field	Description	Action
Collapse icon 	You can hide the resource sections by clicking the Collapse icon  .	Click the icon to toggle the display to show or hide the resource list in the Virtual Services section.
	 Note: When the section is hidden, you can view the number that is displayed of the resources that are associated.	
Show all icon 	You can opt to view the virtual services that are associated with the component either on the Resources or Instances page.	Perform the following steps: <ul style="list-style-type: none"> a. Click the Show all icon . b. Select any of the following actions:

Field	Description	Action
	<p>You can start an instance of the virtual service from the Resources page while you can stop a running instance from the Instances page.</p>	<ul style="list-style-type: none"> ▪ To start an instance of the virtual service: <ol style="list-style-type: none"> i. Select Show all in resources page to go to the Resources page. ii. Click the Execute icon  in the row of the virtual service. iii. Configure the settings for the run of the virtual service, if you want to change any of the settings, else click Execute. <p>The virtual service starts to run.</p> ▪ To stop a running instance of the virtual service: <ol style="list-style-type: none"> i. Select Show all in instances page to go to the Instances page. ii. Identify the instance that you want to stop and click the Stop icon  in the Actions column of the selected virtual service instance. iii. Click Ok in the Stop virtual service instance dialog that is displayed. <p>The running virtual service instance is stopped.</p>
Name	<p>You can view the name of the associated virtual service and the path in the repository.</p>	<p>There is no action to perform.</p>
Instances	<p>You can view the state of the running instance of the virtual service resource. The state is displayed as <code>Running</code> only if the instance is running.</p>	<p>There is no action to perform.</p>
Action	<p>You can go to the Resources page to view the virtual service resource when you click the Show resource icon .</p>	<p>Click the Show resource icon  to view the virtual service on the Resources page.</p> <p>The virtual service resource is displayed on the Resources page.</p>

Field	Description	Action
	You can start an instance of the virtual service from the Resources page.	
Expand icon 	<p>You can expand the virtual service card only if the virtual service is a running instance.</p> <p>In the expanded card, you can view the details such as the Environment, the total number of requests received by the virtual service, and the relative time from when the virtual service had started.</p> <p>You can stop a running instance of the virtual service from the Instances page.</p>	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Expand the virtual service card for a running instance. <p>You can view the details of the instance.</p> <ol style="list-style-type: none"> Click the Show instance icon  to go to the Instances page to view the running instance of the virtual service. <p>The running instance of the virtual service is displayed on the Instances page.</p>

Results

You have viewed the resources that are associated with a component from the **System Model** page.

Removing resources that are associated with components

When you do not want to retain the resource that you associated with a component in HCL DevOps Test Hub (Test Hub), you can delete the resource association from the system model.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).
- Associated resources in a project with components in the system model. See [Associating resources with components on page 359](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

4. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.

5. Click the component that you associated with the resources in the project repository.

The resources associated with the component are displayed under the resource type in the **Associate** tab.

6. Select the resource that you want to remove from the component by clicking the checkbox of the resource.

You can select any or all resources for removal.

7. Click the **Delete selected** icon  in the row of the resource type.

The **Disassociate...** dialog is displayed for the type of resource that you selected. For example, the **Disassociate Suites and Tests** dialog is displayed when you selected the test assets.

8. Click **Delete**.

The **Edit branch** dialog is displayed.

9. Select your action that depends on the existence of an *Edit* branch from the following options:

- Select an existing *Edit* branch that you created previously from the list of branches listed in the **Select branch** field, and then click **Save**.
- Click **Create new branch**, enter a name for the branch, and then click **Save**, if you want to create a new branch.

The dialog that is displayed depends on the type of resource you selected and can be any of the following types:

- The **Dissociate suites and tests** dialog is displayed, if you selected Suites or tests as the resources to dissociate.
- The **Dissociate virtual services** dialog is displayed, if you selected the virtual services as the resources to dissociate.

10. Click **Delete**.

Results

You have removed resources that were associated with components in the system model.

What to do next

You must publish the dissociation changes in the system model to the project repository if you want the changes to be reflected in the system model for all members of the project. See [Publishing changes of associating or dissociating resources with components on page 367](#).

Publishing changes of associating or dissociating resources with components

After you associate or dissociate resources in a project with a component in the system model in HCL DevOps Test Hub (Test Hub), you must publish the changes made to the branch in the project repository so that the association changes are visible to the other members of the project.

Before you begin

You must have completed the following tasks:

- Created and published the system model. See [Creating a system model on page 347](#).
- Created components in the system model. See [Creating components on page 349](#).
- Associated resources in a project with components in the system model. See [Associating resources with components on page 359](#).
- Dissociated resources from components in the system model. See [Removing resources that are associated with components on page 365](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.



The **Active projects** page is displayed.

3. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

4. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.

5. View the number of uncommitted changes made to the system model that is displayed along with the  icon.
6. Click the  icon.

You can view the options for unpublished changes made to the components and associations in the system model. The options available for association changes are as follows:

- Review changes
 - Discard changes
 - Publish changes
7. Perform any of the following actions to publish the association or dissociation changes:
 - Review the unpublished changes before you publish them. Go to [Step 8 on page 367](#).
 - Publish the changes. Go to [Step 9 on page 368](#).
 8. Perform the following steps to review the unpublished changes:

- a. Select the **Review changes** option.

The **Changes** page is displayed. You can find the following information about the changes that you made to the system model when you associated resources with components or deleted the association:

- The *Edit* branch that you selected to contain the changes is displayed in the **Branch** field. You can select the *Edit* branch that you used before associating resources to components, if it is not already displayed as selected.
- The projects that contained the resources, which you associated with components in the system model are listed under the **Asset name** column.
- The type of change is listed under the **Changes** column.
- The name of the member who made the change is displayed under the **Last changed by** column.
- The relative time when the change was made is displayed under the **Last updated** column.
- The number of changes that the *Edit* branch is ahead of from the project repository branch is indicated by the **commits ahead** option. You can view the details by clicking the **commits ahead** option and in the dialog that displays, you can discard the pending commits by clicking the **Reset** option.
- The number of changes that the *Edit* branch is behind from the project repository branch is indicated by the **commits behind** option. The *Edit* branch can be behind the project repository branch if another member has published or committed other changes in the system model to the same project repository branch. You can view the details by clicking the **commits behind** option. You can merge all the changes to your *Edit* branch so that the *Edit* branch is updated with all the changes made in the project repository branch.

- b. Click **Publish changes**.

- c. Enter a commit message in the **Description of change** field, and then click **Publish**.



Note: If you do not want to commit the changes that you made to the system model, you can select the **Discard changes** option.

9. Perform the following steps to publish the unpublished changes:

- a. Click the **Publish changes** option.

- b. Enter a commit message in the **Description of change** field, and then click **Publish**.



Note: If you do not want to commit the changes that you made to the system model, you can select the **Discard changes** option.

Results

You have published the association changes that you made in the system model to the repository in the project. Other members of the team space can view the association changes and use the system model.

Viewing the system model

You can view an existing system model that is in a team space in HCL DevOps Test Hub (Test Hub) if you are a member of the team space.

Before you begin

You must be a licensed user of Test Hub and a member of a team space.

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.

Results

You viewed an existing system model in the team space from the **System Model** page.

Deleting a system model

When you want to remove an existing system model to create a new system model in HCL DevOps Test Hub (Test Hub), you must delete the existing system model because only one system model is supported in a team space.

Before you begin

You must have ensured that a system model exists in the team space.

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.

The **Active projects** page is displayed.

3. Click **System Model** on the left navigation pane.

The **System Model** page is displayed.

4. Click the **Open action menu** icon  in the right pane.
5. Click the **Delete** icon .

The **Delete System Model** dialog is displayed.

6. Click **Delete**.

Results

You deleted the system model that existed in the team space.

What to do next

You must publish the changes to the team space repository. See [Publishing changes to the system model on page 359](#).

Deleting a repository that is added to a team space

You can delete a repository that you added to the team space in HCL DevOps Test Hub (Test Hub) when you do not want to use a team space repository.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* or an *Architect* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Added a repository to the team space. See [Adding a repository to a team space on page 345](#).

1. Log in to Test Hub, if you are not already logged in.
2. Open the team space to which a team space repository is added.


The **Active projects** page is displayed.

3. Click the **Settings** icon  from the left navigation pane, and then select **Manage team space**.

The **Details** page is displayed.

4. Click **Repository** on the left navigation pane.

The **Repository** page is displayed.


5. Click the **Open action menu** icon .
6. Click **Delete** in the action menu list.

The **Delete repository** dialog is displayed.

7. Select the **I understand...** checkbox, and then click **Delete**.

Results

You have deleted the team space repository.

 **Restriction:** When you delete the team space repository, the system model if it exists is also deleted, and you or other members of the team space cannot add a system model.

Prerequisites to running tests

Before you can run a test asset in a project on HCL DevOps Test Hub (Test Hub), you must read the information about the different tests assets. You can add a remote static agent, Docker host, or Kubernetes cluster to the project and then select it as an alternate location to run the test assets.

You can find the following information about the prerequisite tasks:

- [Test run considerations for AFT Suites or Functional tests on page 371](#)
- [Test run considerations for API Suites or API tests on page 373](#)
- [Test run considerations for browser-based Web UI tests on page 376](#)
- [Test run considerations for JMeter tests on page 376](#)
- [Test run considerations for JUnit tests on page 378](#)
- [Test run considerations for Postman tests on page 380](#)
- [Test run considerations for schedules on page 382](#)
- [Test run considerations for Selenium tests on page 383](#)
- [Test run considerations for using mobile device clouds on page 384](#)
- [Management of agents on page 386](#)
 - [Test run considerations for running tests on remote performance agents on page 387](#)
- [Management of Docker hosts on page 401](#)
 - [Test run considerations for running tests on remote Docker hosts on page 401](#)
- [Management of remote Kubernetes clusters on page 417](#)
 - [Considerations for using a remote Kubernetes cluster on page 417](#)
- [Management of mobile device cloud credentials on page 432](#)

Related information

[Test run configurations on page 446](#)

Test run considerations for AFT Suites or Functional tests

Before you configure an AFT Suite or a Functional test run in HCL DevOps Test Hub (Test Hub), you must first read the considerations that you need to take into account.

When you want to run an AFT Suite or Functional test in a project on Test Hub, you can check if the AFT Suite or Functional test meets any of the following conditions:

If...	Then...
The AFT Suite or Functional test uses the settings configured in an AFT XML file for a remote agent	<p>You must ensure that you have configured the location element in the AFT XML to point to a remote agent.</p> <p>For more information about setting the location element, refer to Using an XML file to run multiple Web UI tests and Compound Tests simultaneously.</p>
You want to run an AFT Suite or Functional test on a remote agent	<p>You must run the test on the remote agent before you commit the test asset to the remote repository.</p> <p>When you run the test on the remote agent, the following events occur:</p> <ul style="list-style-type: none"> • The agent is added as the location in the AFT XML file, on which the test is to be run. • The agent is displayed as the agent on which the test is to be run, under the Host column within the Location tab in the Execute test asset dialog.
You commit the test asset to the remote repository without running the test on the remote agent	The agent is not displayed under the Host column within the Location tab in the Execute test asset dialog.

Before you configure the run for an AFT Suite or a Functional test, you must complete the following tasks, if they are applicable:

- Add the remote agents, on which the test is to be run, to the project on Test Hub. See [Adding a performance agent to a project on page 389](#).
- After you add the agent, you can select the agent configured in the AFT XML file as the location for the test run or select the remote agents that you add to the project, when you configure a run for an AFT Suite.

The remote agents are shown as the agents available for selection under the **Override** column within the **Location** tab in the **Execute test asset** dialog.

You can then select the remote agent as the location to run the AFT Suite or Functional test when you are configuring the test run. Alternatively, you can enter the argument `-swaplocation <configured_agent_location>:<overriding_agent_location>` in the **Program Arguments** field in the **Advanced Setting** dialog when you are configuring the test run.

For running an AFT Suite, see [Configuring an AFT Suite run on page 453](#).

Related information

[Test run configurations on page 446](#)

Test run considerations for API Suites or API tests

Before you configure an API Suite or API test run, you must first read the considerations that you need to take into account.

You can find the following information about API Suites:

- [Important information about API Suites or tests on page 373](#)
- [Important information about report configuration in API Suites or tests on page 375](#)

Important information about API Suites or tests

When you want to run an API Suite or test from a project on HCL DevOps Test Hub (Test Hub) and the test asset in HCL DevOps Test Integrations and APIs (Test Integrations and APIs) meets any of the following conditions:

- The Suite or test refers to local stubs that use a transport other than HTTP or MQ.
- The Suite or test refers to local stubs that use the HTTP transport.
- The transport used in the tests or Suite is configured with the hostname set to *localhost* for the HTTP/TCP proxy.
- The Suite or test has tests that use a transport and the transport requires third-party application `JAR` files for a successful run.

You must then refer to the following table for the next steps:

If...	Then...
The test asset refers to local stubs that use a transport other than HTTP or MQ.	You must perform any of the following actions before you commit the test asset to the remote repository: <ul style="list-style-type: none"> • Remove the reference to the local stub from the test asset. • Publish the stubs to Test Virtualization Control Panel. To publish and edit stubs, see Publishing stubs .
The tests refer to local stubs that use the HTTP transport.	You must perform the following actions before you commit the test asset to the remote repository:

If...	Then...
	<ol style="list-style-type: none"> 1. Create a test or test Suite that contains the stub and tests that run on the stub. 2. Configure the physical HTTP transport with the following parameters in the Client tab in the Physical View: <ul style="list-style-type: none"> ◦ Set the <code>hostname</code> or IP address of the HTTP Proxy server in the Proxy Host field. ◦ Set <code>3128</code> as the proxy server port in the Proxy Port field. 3. Set the reference to the stub for the tests in the test suite scenario from the Scenario Editor dialog from the Test Factory view by selecting the <i>Live system</i> option in the Satisfied by column to run the stubs on the HTTP proxy registered with Test Hub. 4. Save the project.
<p>The transport used in the Suite or tests is configured with the hostname set to <i>localhost</i> for the HTTP/TCP proxy.</p>	<p>You must replace the hostname with <i>fully-qualified-domain-name</i> or <i>IPAddress</i> of the proxy host in the tests before you commit the test asset to the remote repository.</p>
<p>You plan to run the API Suite or test on the computer where Test Hub is installed and the API Suite or test that use a transport from any of the following third-party applications:</p> <ul style="list-style-type: none"> • Camel • CentraSite • CICS • Coherence • Database • IMS • Integra • JMS • SAP RFC • Software AG Universal Messaging • TIBCO EMS • TIBCO Rendezvous 	<p>You must complete the following tasks:</p> <ol style="list-style-type: none"> 1. Identify the location of the third-party application JAR files. You can use Library Manager to know the location where the third-party application JAR files are saved. 2. Copy the third-party application JAR files to the computer where Test Hub is installed. <p>See Copying third-party application Jars to Kubernetes on page 160.</p>

If...	Then...
<ul style="list-style-type: none"> • TIBCO SmartSockets • WebSphere Application Server Service Integration Bus (SiBus) • WebLogic • Software AG webMethods • WebSphere MQ 	
<p>You plan to run the API Suite or test on a remote Docker host and the API Suite or test contains tests that use a transport from any of the following third-party applications:</p> <ul style="list-style-type: none"> • Camel • CentraSite • CICS • Coherence • Database • IMS • Integra • JMS • SAP RFC • Software AG Universal Messaging • TIBCO EMS • TIBCO Rendezvous • TIBCO SmartSockets • WebSphere Application Server Service Integration Bus (SiBus) • WebLogic • Software AG webMethods • WebSphere MQ 	<p>You must complete the following tasks:</p> <ol style="list-style-type: none"> 1. Identify the location of the third-party application JAR files. You can use Library Manager to know the location where the third-party application JAR files are saved. 2. Copy the third-party application JAR files to the computer where the remote Docker host is installed. <p>See Copying third-party application Jars to a remote Docker host on page 406.</p>

Important information about report configuration in API Suites or tests

You must configure test Suites or tests in Test Integrations and APIs to use a results database so that the details of the test results can be captured and displayed in the Test Hub API results reports. See [Configuring the project results database](#).

If your server projects use either a Microsoft SQL Server, MYSQL, or both as a project results database, you must copy the Microsoft SQL Server and MySQL database JAR files to the JDBC folder in the path `/data/JDBC` on Test Hub. See [Copying third-party application Jars to Kubernetes on page 160](#).

The database JAR files that you need are based on the version of Microsoft SQL Server or MSSQL that you use with Test Integrations and APIs. See [Adding Microsoft SQL Server and MySQL drivers](#).

To run an API Suite, see [Configuring an API Suite or an API test run on page 486](#).

Test run considerations for browser-based Web UI tests

Before you configure a Web UI test or a Compound Test that contains Web UI tests, which use either the Chrome or Edge browsers, you must first read the considerations that you need to take into account.

When you want to run a Web UI test or a Compound Test that contains Web UI tests, which use either the Chrome or Edge browser from HCL DevOps Test Hub (Test Hub), you must consider the following conditions:

- You cannot run these tests on a remote Docker host. You must not select the remote Docker host as a location to run these tests when you configure a test run.
- You must perform any of the following actions:
 - Accept the license agreement of the built-in Chrome and Edge browsers at the time of signing in or logging in to Test Hub.
 - Accept the license agreement of the built-in Chrome and Edge browsers, if it is not accepted at the time of signing in or logging in. See [Managing third-party licenses on page 155](#).

Test run considerations for JMeter tests

Before you can run JMeter tests on HCL DevOps Test Hub (Test Hub), you must read the considerations that you must take into account and complete the tasks indicated.

Before you run JMeter tests that you create in JMeter, you must have completed the following tasks:

- You must ensure that the JMeter extension on Test Hub is enabled. You must enable the JMeter extension either at the time of installation of Test Hub or before you add the repository that contains JMeter tests to the server project. See [Enabling server extensions on page 166](#).
- You must have committed the following JMeter assets or resources as a JMeter project to a remote Git repository:



Note: You must have created a project directory and added the associated tests or assets in sub directories, that might be required for the JMeter tests to run successfully.

- The `jmeterRoot.jprj` project file in the project directory.
- The JMeter tests in a sub directory with a name you provide within the project directory.
- Optionally, the Java Key Store that is required at test run time in the `certs` sub directory.
- Optionally, the JMeter properties files that are required at test run time in the `properties` sub directory.
- Optionally, the library jar files that are required at test run time in the `lib` sub directory.
- Optionally, the test dependencies that are required at test run time in the `deps` sub directory.

For example, the project directory can be *myproject* that contains the `jmeterRoot.jprj` project file. In the project directory, you can create the following sub directories:

- The `mytests` sub directory that contains the JMeter tests.
- Optionally, the `certs` sub directory that contains the Java Key Store.

For example, the path to the key store can be specified in the `system.properties` file as follows:

```
javax.net.ssl.trustStore=$(deploy.home)/certs/simpleStore.jks
```

Where `<simpleStore.jks>` is the Java Key Store and the `<deploy.home>` is the keyword to access the key store at the test run time.

- Optionally, the `properties` sub directory that contains properties files such as `system.properties`, `jmeter.properties`, `reportGenerator.properties`, or `user.properties`.
- Optionally, the `lib` sub directory that contains the **lib** and **lib/ext** content to be used at test run time.

For example, the **amq client** and the **plugins** required to parse the **jmx** that contains **amqp** calls.

- Optionally, the `deps` sub directory that contains the test dependencies.

The resulting project directory can be as follows:

```
myproject (that contains the jmeterRoot.jprj)
myproject/mytests
myproject/certs
myproject/properties
myproject/lib
myproject/deps
```

- You must ensure that you use a relative path when JMeter tests refer to other test assets. For example, if the JMeter test that is in the `myproject/mytests` folder refers to a `mycsv.csv` file that is in the `myproject/deps` folder, then the `mycsv.csv` is referred as follows: `../deps/mycsv.csv` in the JMeter test.
- You can create the JMeter project file named as `jmeterRoot.jprj`.


The project file is a java properties file. The JMeter project file is required so that Test Hub can identify the asset as a JMeter project. You can specify additional information in the project file. You can specify the names of the sub directories created and through keywords you can specify the path in the properties files whether they are relative to the root directory or to the tests sub directory.

For example, if you have created the following sub directories in the project directory called *myproject* that contains the `jmeterRoot.jprj` project file:

- `myproject/mytests`
- `myproject/properties`

If the JMeter tests are contained in the following path: `myproject/mytests/testplan/test1.jmx`, and some dependencies that are also required to be deployed at test run time exists in the following path: `myproject/mytests/testplan/localdir`. You must specify these file paths in the project file as follows:

```
testDir.localtest.sub.dir=localdir  
aSecondRootDir.sub.dir=dir2  
athirdRootDir.sub.dir=dir3
```

 **Restriction:** You cannot run the test assets on a remote Docker host.

Related reference

[Troubleshooting issues on page 1077](#)

Related information

[Configuring a run of a JMeter test on page 665](#)

Test run considerations for JUnit tests


Before you can run JUnit tests on HCL DevOps Test Hub (Test Hub), you must read the considerations that you must take into account and complete the tasks indicated.

Before you run JUnit tests, you must have completed the following tasks:

- You must ensure that the JUnit extension on Test Hub is enabled. You must enable the JUnit extension at the time of installation of the server software or before you add the repository that contains the JUnit tests to the server project. See [Enabling server extensions on page 166](#).
- You must have created the JUnit tests that use JDK 8, as part of a Maven V3.6.3 project. The Maven project must contain the Maven Surefire plugin, which is required to run the JUnit tests contained in the Maven project. The Maven project as a `pom.xml` must be committed to the remote repository.



Note: If you create the JUnit tests on other versions of JDK, you must copy the JDK version to the `userlibs` pod under the `/data/junit-ext/jdks` directory. See [Copying third-party application Jars to Kubernetes on page 160](#).

 **Restriction:** You cannot run the test assets on a remote Docker host.

Related reference

[Troubleshooting issues on page 1077](#)

Related information

[Configuring a run of a JUnit test on page 674](#)


Test run considerations for using Machine learning capabilities

Before you configure a Performance Test or Schedule run to use Machine learning (ML) capabilities, you must first read the considerations that you need to take into account.

When you want HCL DevOps Test Hub (Test Hub) to analyze Performance tests or Schedules by using ML capabilities, you must ensure that Performance tests or Schedules conform to the following criteria:

- Must contain HTML tests.
- Must contain stress or performance user profiles.
- Must contain the following minimum stress test requirements:
 - 20 users or more.
 - A minimum of three different user groups and each group has an activity of more than two minutes.
 - Stages that have a reasonably high number of samples within each stage.
 - Stages do not have an increasing number of users.
 - Tests run for longer durations so that a trend can be observed.

You can refer to the following table to view the criteria for analysis of the parameters:

Parameter analyzed	Criteria for analysis
Response Time Lock-Step Pattern	<p>The analyzer identifies the Response Time Lock-Step Pattern parameter in the overall page response time observed against the user count based on the following criteria:</p> <ul style="list-style-type: none"> • Tests have a minimum of 20 users. • Tests run for longer durations for a trend to be observed. • Tests contain a minimum of three different user groups and each group has an activity of more than two minutes.
Response Time Standard Deviation Pattern	<p>The analyzer attempts to detect the response time of pages that are more than thrice the value of the standard deviation calculated for the page response time as the Response Time Standard Deviation Pattern parameter.</p>
Throughput Drop Pattern	<p>The analyzer attempts to detect sudden drops in network throughput as the Throughput Drop Pattern parameter that is based on the following criteria:</p> <p> Note: Sudden drops in throughput might be related to the performance tool itself, issues with network connectivity, or issues with signal-scalability of the system under test.</p> <ul style="list-style-type: none"> • Tests have a minimum of 20 users. • Tests have stages with a reasonably high number of samples within a stage. • The length and intensity of the throughput drops that are less than 20% of the median observed within a stage are considered by the analyzer. When no

Parameter analyzed	Criteria for analysis
	<p>throughput is received within 60 seconds, such drops are considered in the analysis. Short shark-tooth patterns in drops are not considered by the analyzer.</p> <ul style="list-style-type: none"> • Parts of the stage with increasing users are not considered for analysis. <p>For example, when the test adds one user every minute and the user range is between one user to 500 users, the Throughput Drop Pattern parameter is not analyzed.</p>

Related information

[Machine learning capabilities in DevOps Test Hub on page 826](#)


[Test run configurations on page 446](#)

Test run considerations for Postman tests

Before you can run Postman tests on HCL DevOps Test Hub (Test Hub), you must read the considerations that you must take into account and complete the tasks indicated.

Before you run Postman tests that you create in Postman, you must have completed the following tasks:

- You must ensure that the Postman extension on Test Hub is enabled. You must enable the Postman extension at the time of installation of the server software or before you add the repository that contains Postman collections to the server project. See [Enabling server extensions on page 166](#).

 **Restriction:** If you add a Git repository with Postman collections to a server project before you enable the Postman microservice, the Postman collections might not be displayed in the **Execution** page for you.

- You must have exported the following resources from Postman and saved the resources on your computer:
 - Collections and their associated variables and environments from Postman in which all collections are exported as a single file.



Note: You can export as many collections as you want.

For more information, refer to [Exporting a collection from Postman](#) and [Exporting data dumps](#).

- Environments configured in Postman, if you are exporting only the environments as separate JSON files.

For more information, refer to [Exporting environments from Postman](#).

- You must have committed the following Postman assets or resources to the remote repository:
 - Collections and their associated variables and environments that you exported. You can export all resources that are in your workspace in Postman. You can also export the collections along with the variables as a single file and the environments separately as another file.

For example, you can save the exported file as `MyCollectionsVarEnv.json` that contains the collections, variables, and environments or you can export only the collections and variables as `MyCollectionsVar.json`.

- Environments that you exported, if you are exporting only the environments as separate JSON files. For example, you can save the exported file as `MyEnv.json`.
- An empty text file with the `.postman` extension that is contained in the same directory as the other Postman resources, so that the contents in the directory are attributed as Postman resources. For example, you can create `MyProject.postman`.



Note: The `.postman` file enables Test Hub to identify the test resources as Postman resources.

For example, if you have saved the Postman resources that you downloaded from Postman as follows:

- `MyCollectionsVarEnv.json` or `MyCollectionsVar.json` that contains the Postman collections.
- `MyEnv.json` that contains the environments associated with the Postman collection.

The project directory can be as follows:

```
/Project
/Project/MyProject.postman
/Project/MyCollectionsVarEnv.json
```

Alternatively, the directory can be as follows:

```
/Project
/Project/MyProject.postman
/Project/MyCollectionsVar.json
/Project/MyEnv.json
```



Restriction: You cannot run the test assets on a remote Docker host.

Related reference

[Troubleshooting issues on page 1077](#)

Related information

[Configuring a run of a Postman test on page 683](#)

Test run considerations for schedules

Before you configure a Rate Schedule or VU Schedule run, you must first read the considerations that you need to take into account.

Important information about JMeter tests in schedules

When you want to run JMeter tests as part of the VU Schedule or Rate Schedule on HCL DevOps Test Hub (Test Hub), you must have completed the following tasks:

- Installed the HCL DevOps Test Performance Agent on a computer.
- Installed the JMeter application on the computer where you have installed the HCL DevOps Test Performance Agent.
- Set the environment variable **JMETER_HOME** that points to the JMeter installation directory.

Important information about Resource Monitoring sources in schedules

When you want to run schedules on Test Hub, you must have completed the following tasks in Test Performance:

- Created a performance schedule of **Rate** or **VU** type.
- Added Resource Monitoring sources, or the labels of Resource Monitoring sources to a schedule. Refer to [Adding resource monitoring sources to a performance schedule by using labels](#).


Before you configure a run of a Schedule, you must complete the following tasks in Test Hub:

- Add Resource Monitoring sources to your project in Test Hub. See [Monitoring an NGINX server on page 847](#).
- Enable Test Hub as Resource Monitoring service only if you want to replace Resource Monitoring labels set in Test Performance with the labels that you created in Test Hub. See [Controlling resource monitoring sources in a schedule on page 878](#).
- Perform the following steps when you want to override the Resource Monitoring labels in the test asset with the Resource Monitoring labels that you create in Test Hub:
 1. Click the **Resource Monitoring** tab.



Note:

The **Resource Monitoring** tab displays only if you enabled the **Resource Monitoring from Service** option in Test Performance when you created the test asset.

2. Click , and press the `Ctrl + Space bar` keys, or enter the initial letter of a label to select a label in the list.

**Note:**

You can select or add labels only if you added the labels in the **Resource Monitoring Sources** page in Test Hub.

For running a Rate Schedule or VU Schedule, see [Configuring a run of a Rate Schedule or VU Schedule on page 692](#).

Related information

[Test run configurations on page 446](#)

Test run considerations for Selenium tests

Before you can run Selenium tests on HCL DevOps Test Hub (Test Hub), you must read the considerations that you must take into account and complete the tasks indicated.

Before you run Selenium tests, you must have completed the following tasks:

- You must ensure that the JUnit extension on Test Hub is enabled. You must enable the JUnit extension at the time of installation of the server software or before you add the repository that contains the Selenium tests to the server project. See [Enabling server extensions on page 166](#).
- You must have created the Selenium tests that use JDK 8, as part of a Maven V3.6.3 project. The Maven project must contain the Maven Surefire plugin, which is required to run the Selenium tests contained in the Maven project. The Maven project as a `pom.xml` must be committed to a remote Git repository.



Note: If you create the Selenium tests on other versions of JDK, you must copy the JDK version to the `userlibs` pod under the `/data/junit-ext/jdks` directory. See [Copying third-party application Jars to Kubernetes on page 160](#).

- You must ensure that the following flags are configured in the Selenium test:
 - Firefox is configured as the web browser to run the test.
 - Firefox is enabled to accept all certificates (secured and unsecured).
 - Firefox is configured to not display alerting messages when the test runs in the Firefox browser.
- You must set the system property to point to the Firefox driver (`linux_x64_geckodriver`) that is provided along with the JUnit server extension and is available in the `/tmp` folder.

For example,

```
System.setProperty("webdriver.gecko.driver", "/tmp/linux_x64_geckodriver")
```



Restriction: You cannot run the test assets on a remote Docker host.

Related reference

[Troubleshooting issues on page 1077](#)

Related information



[Configuring a run of a Selenium test on page 704](#)

Test run considerations for using mobile device clouds

Before you configure runs of tests that use mobile device clouds as the location, you must first read the considerations that you need to take into account.

Before you can use mobile devices that are connected to a mobile device cloud from HCL DevOps Test Hub (Test Hub), you must complete the following tasks:

- Create any of the following types of tests that are supported to be run from Test Hub.

Test type	Description
AFT Suite	<p>The AFT Suite must specify the path to the tests, device names, and the device cloud location to run the tests.</p> <p>AFT Suites that specify multiple tests or devices on the same device cloud are supported.</p> <p> Restriction: AFT Suites that specify a single test on multiple devices or multiple tests on multiple devices when the devices are connected to multiple device clouds are not supported.</p>
Compound Tests that contain mobile tests or Web UI tests that use a device cloud	<p>The Compound Tests must specify multiple mobile tests or multiple Web UI tests that run on a device connected to a single device cloud.</p> <p> Restriction: Compound Tests that specify tests on single or multiple devices that are connected to multiple device clouds are not supported.</p>
Individual mobile test or Web UI test that use a device cloud	<p>The individual test must be either a mobile test or a Web UI test that runs on a device connected to a single device cloud.</p>

- Ensure that you have valid credentials for the mobile device clouds, which are supported to be connected from Test Hub. You must have the details as listed in the following table for the mobile device cloud that you want to connect and use:

Attribute	Description	Mobile device cloud			
		BitBar Cloud	BrowserStack Cloud	pCloudy Cloud	Perfecto Mobile Cloud
Host	The hostname of the mobile device cloud instance.	Required	Required	Required	Required
API key	The API key of your cloud account to authenticate the connection.	Required	Required	Required	Not applicable
Security token	The token used to authenticate the connection.	Not applicable	Not applicable	Not applicable	Required
Project	The project name to use on the mobile device cloud.	Required	Required	Not applicable	Not applicable
Username	The username of your cloud account.	Not applicable	Required	Required	Not applicable

- Complete the following prerequisite tasks to enable the devices on the mobile device cloud for testing mobile apps or mobile web browsers:

Device cloud	Testing on Android OS	Testing on iOS
BitBar Cloud	<ul style="list-style-type: none"> ◦ Uploaded the <code>.apk</code> file of the Android application that you want to test. ◦ Installed the Android application that you want to test on the Android device. 	<ul style="list-style-type: none"> ◦ Installed the <code>.ipa</code> or <code>.app</code> file of the iOS application that you want to test. ◦ Installed the iOS application that you want to test on the iOS device.
BrowserStack Cloud		
Perfecto Mobile Cloud		
pCloudy Cloud	<ul style="list-style-type: none"> ◦ Uploaded the <code>.apk</code> file of the Android application that you want to test. ◦ Installed the Android application that you want to test on the Android device. 	<ul style="list-style-type: none"> ◦ Uploaded the <code>.ipa</code> file of the iOS application that you want to test. ◦ Installed the iOS application that you want to test on the iOS device.

Device cloud	Testing on Android OS	Testing on iOS
	<ul style="list-style-type: none"> ◦ Verified that the device you want to test is available and in the <i>Released</i> state on the pCloudy Cloud. 	<ul style="list-style-type: none"> ◦ Verified that the device you want to test is available and in the <i>Released</i> state on the pCloudy Cloud.

- Ensure that as a *Project Owner* or *Tester*, you are an authorized user of the cloud credential in your team space. See [Viewing details of cloud credentials on page 439](#).
- Ensure that you create a variables file that contains the details of the agent or device cloud, if you do not want to use the cloud credentials on the server.

Management of agents

You can find information about managing performance agents or API agents that you install on remote computers and then configure the remote agents to register with HCL DevOps Test Hub (Test Hub). After the agents are registered, you can add them to your projects and then select them as locations to run tests or virtual services.

When you plan to run any of the tests from a project on Test Hub on a remote performance agent, see [Management of performance agents on page 386](#).

When you plan to run any of the supported virtual services from a project on Test Hub on a remote API agent, see [Management of API agents on page 394](#).

Management of performance agents

You can manage the performance agents that you want to use as locations to run tests from HCL DevOps Test Hub (Test Hub). When you install the agents by using your offline user token, the agents are registered with the team space in Test Hub. You can view the registered agents before you add them to your projects. You can then select them as locations to run tests.

When you plan to run any of the tests on a remote performance agent from a project in Test Hub, you must have completed the following tasks:

- Installed the performance agent of the same version as the version of Test Hub. Refer to [Installing DevOps Test Performance Agent](#).
- Configured the performance agent with your offline user token so that the performance agent can register with Test Hub.



Note: When you configure the performance agent by using your offline user token, Test Hub considers you as the owner of the performance agent.

- Verified that you have write permissions to the `majordomo.config` file so that certain transactions can be written by Test Hub to the `majordomo.config` file.



Note: You can verify if you have the *Read* and *Write* permissions by checking the properties of the `majordomo.config` file.

- Viewed the performance agents that are registered with Test Hub. See [Viewing performance agents that are registered with DevOps Test Hub on page 388](#).
- Added the remote performance agents that are registered with Test Hub to your project on Test Hub. See [Adding a performance agent to a project on page 389](#).
- After you add the remote performance agents to your project, Test Hub determines the capabilities that are provided by the remote performance agent and displays the details and capabilities of the performance agent on the **Agents and Intercepts** page. You can also define the capabilities that you want to add, and then view, edit, or delete the capabilities that you defined for the performance agents in your project. See [Working with the capabilities of a performance agent on page 391](#).

Test run considerations for running tests on remote performance agents

Before you configure a test to run on a remote performance agent from HCL DevOps Test Hub (Test Hub), you must read the considerations that you must take into account.

Supported tests

You can run the following tests that are supported to be run on a remote performance agent:

- AFT Suite
- Compound Tests that contains performance tests
- Compound Tests that contains Web UI tests
- Compound Tests that contains traditional HTML tests
- Rate Schedule
- VU Schedule

Prerequisite tasks

Before you can run any of the supported tests on a remote performance agent from a project on Test Hub, you must have completed the following tasks:

- Installed the HCL DevOps Test Performance agent on the remote computer. See [Management of agents on page 386](#).
- Configured the agent with your offline user token so that the agent can connect to Test Hub.
- Added the remote agents to your project on Test Hub. See [Adding a performance agent to a project on page 389](#).



Note: If you want to run an AFT Suite (that contains Web UI tests) on a performance agent, you must start the performance agent as a non-admin user so that certain browsers that are configured can start correctly during the test run time.

Before you add the tests to the remote repository, you must have completed the following tasks:

- Associated the performance agents with the performance schedule in Test Performance.
- Specified the hostname of the remote workstation in the AFT XML file.

You can continue to configure a run for any of the following supported test types:

- [Configuring an AFT Suite run on page 453](#)
- [Configuring a run of a Compound Test that contains performance tests on page 569](#)
- [Configuring a run of a Compound Test that contains Web UI tests on page 533](#)
- [Configuring a run of a Compound Test that contains HTML tests on page 498](#)
- [Configuring a run of a Rate Schedule or VU Schedule on page 692](#)

Viewing performance agents that are registered with DevOps Test Hub

After you install the remote performance agents and configure them with your offline user token, the performance agents register with HCL DevOps Test Hub (Test Hub) and you can view the registered performance agents on the **Agents and Intercepts** page.

Before you begin

You must have completed the following tasks:

- Installed the performance agent of the same version as the version of Test Hub.
- Configured the performance agent with your offline user token so that the performance agent can register with Test Hub.



Note: When you configure the performance agent to connect to the server, you must specify the offline user token that you generated on Test Hub. This token defines your ownership over the agent.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane.

The **Agents and Intercepts** page is displayed.

You can view the API agents, Dockers, intercepts, Kubernetes clusters, or performance agents that are registered with Test Hub.

3. You can view the agents that you own in any of the following ways:

- Search for the agent by entering the name of the agent in the **Search** field.



Note: You can enter either the full name or any text that is in the name. The search is enabled for case sensitive text that you can enter.

- Sort the **Type** column to sort the items and display the agents in the rows at the top of the table.
- Sort the **Agents** column and identify the agents by their names or by their owner.

You can view the following details about the agents that are registered with Test Hub:

- The projects to which the agent is added, are displayed in the **Projects** column.
- The status of the agent is displayed in the **Status** column.



Note: You can add agents that are in the `Ready` state to your project, and then use them as locations to run tests or virtual services.

Results

You have viewed the registered agents from the **Agents and Intercepts** page.

What to do next

You can perform any of the following tasks:

- You can add the performance agents that you own to your project. See [Adding a performance agent to a project on page 389](#).
- You can add the API agents that you own to your project. See [Adding a performance agent to a project on page 389](#).

Adding a performance agent to a project

If you have performance tests that distribute workload across different workstations or Accelerated Functional Testing (AFT) Suites that need to run on different platforms and browsers, you can add HCL DevOps Test Performance Agents to your projects on HCL DevOps Test Hub (Test Hub) to run tests on remote workstations.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Installed the agent by using your offline user token that makes you the owner of the agent or you are assigned the role of a *Team Space Owner*.

About this task

Only you can add or remove a performance agent that you own from the project. However, any member of your project with the permission to run the tests can initiate the run of test assets that are associated with the performance agent.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open your project, and then click **Infrastructure > Agents and Intercepts**.

The **Agents and Intercepts** page is displayed.

3. Click **Add > Add performance agent**.

The performance agents that you configured with your user token are displayed.



Note: You can add the performance agents that you own and that are in the *Ready* state. You can add any number of performance agents to your project.

4. Select the performance agents that you want to use in the project, and then click **Add**.

The performance agents that you added to the project are displayed.

5. Perform the following steps to view details and capabilities of the performance agents:
 - a. Expand the performance agent to view the details panel.

You can find the following details of the performance agent displayed:

Parameter	Description
Version	The version of the installed performance agent.
Address	The IP address of the computer on which the performance agent is installed.
License type	The license type that is enabled for you to use the performance agent.
Load generation service	The status of the performance agent whether it can be used for load generation service.
Last contact	The time of the last contact with the performance agent by Test Hub.

- b. Expand the capabilities panel to view the system and application capabilities.

You can find the following capabilities of the performance agent displayed:

Capabilities		Description
System capabilities	Architecture	The architecture of the CPU processor of the computer on which the performance agent is installed.
	MAC Address	The MAC address of the computer on which the performance agent is installed.
	Operating system	The operating system of the computer on which the performance agent is installed.
	RAM	The memory that is allotted to RAM in the computer on which the performance agent is installed.
Application capabilities	Browser	The browsers along with their version that are installed on the computer on which the agent is installed.
User Defined capabilities	User Defined	The capabilities that you add for a performance agent. You can also tag a performance agent so that you can identify the performance agent to select as a location.

- c. Check for the status of the performance agent from the **Status** column.



Note: You can add performance agents that are in any state to the project.

Results

You have added remote performance agents to your project and viewed the performance agent capabilities that is displayed for each agent that you own.

What to do next

You can perform any of the following tasks:

- Run the tests on the performance agent that is added to the project from the **Execution** page. See [Test run configurations on page 446](#).
- Define the capabilities that you want to add, and then view, edit, or delete the capabilities that you defined for the performance agents in your project. See [Working with the capabilities of a performance agent on page 391](#).

Working with the capabilities of a performance agent

When you installed performance agents V10.1.2 or later, and added them to your project on HCL DevOps Test Hub (Test Hub), you can view capabilities of the performance agents from the **Infrastructure** page. When you installed

performance agents V10.2.0 or later, you can add, view, edit, or delete properties and capabilities of performance agents from the **User Defined** panel of that performance agent.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Installed the agent by using your offline user token that makes you the owner of the agent or you are assigned the role of a *Team Space Owner*.
- Installed a performance agent. See [Management of agents on page 386](#).
- Added performance agents to your project. See [Adding a performance agent to a project on page 389](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open your project, and then click **Infrastructure > Agents and Intercepts**.

The agents that you added to your project are displayed on the **Agents and Intercepts** page.

3. Expand the performance agent.

The **Details** and **Capabilities** tabs are displayed.






4. Click **Capabilities** to view the capabilities of the performance agent.


5. Click **User Defined**.

The **User Defined Capabilities** dialog is displayed.

6. Complete the steps for the task that you want to perform as listed in the following table:

Task	Action
Adding capabilities	Perform the following actions: <ol style="list-style-type: none"> Click Add new. Enter the name of the capability in the Capability Name field. Perform any of the following actions in the User Defined Category field:

Task	Action
	<ul style="list-style-type: none"> ▪ Select an existing category from the list. ▪ Enter a name of the category for the capability, if the category does not exist. <p> Note: You can also click Add new to add a category.</p> <p>The category for the capability is either selected or created and the fields for the property name and value are displayed.</p> <p> Note: The property for a capability is a <i>name-value</i> pair.</p> <p>d. Enter a name for the property in the Property Name field and its corresponding value in the Property Value field for the capability.</p> <p> Note: You can add multiple properties for the same capability under the same category or under different categories by clicking Add new property, and then by entering the name and value for the property.</p> <p>e. Click Save.</p> <p>The capabilities that you added are displayed.</p>
Viewing capabilities that you added	The User Defined Capabilities panel displays the capabilities that you added to the performance agent.
Editing capabilities that you added	<p>Perform the following actions:</p> <ol style="list-style-type: none"> a. Click the Edit icon  in the row of the capability that you want to edit. <p>The user-defined capability is enabled for editing. You can edit the name of the capability, or the property name or property value.</p> <ol style="list-style-type: none"> b. Edit any of the items that you want for the user-defined capability. c. Click Update. <p>The updated capability of the performance agent is displayed.</p>
Deleting capabilities that you added	<ol style="list-style-type: none"> a. Click the Delete icon  in the row of the capability that you want to delete. b. Click Delete. <p>The capability that you selected is deleted along with all the properties that you added for the capability.</p>

Task	Action
Deleting a property defined for a capability	<ol style="list-style-type: none"> Click the Edit icon  in the row of the capability that you want to edit and remove a defined property. The user-defined capability is enabled for editing. Select and delete the contents in the Property Name field and its corresponding Property Value field. Click Update. <p>The property is removed for the capability.</p>

Results

You have added, viewed, edited, or deleted a property or the user-defined capability of a performance agent.

What to do next

You can run the tests on the performance agent from the **Execution** page. See [Test run configurations on page 446](#).

Management of API agents

You can manage the HCL DevOps Test Integrations and APIs (Test Integrations and APIs) agents that you want to use as locations to run tests from HCL DevOps Test Hub (Test Hub). When you install the agents by using your offline user token, the agents are registered with the team space in Test Hub. You can view the registered agents before you add them to your projects. You can then select the agents as locations to run virtual services.



Note: The API agent is referred interchangeably as either the Test Integrations and APIs Agent or API agent in the documentation.

Before you run any of the supported virtual services on a remote Test Integrations and APIs Agent from a project on Test Hub, you must have completed the following tasks:

- Installed the Test Integrations and APIs Agent of the same version as the version of Test Hub. Refer to [Installing DevOps Test Integrations and APIs Agent](#).
- Configured the Test Integrations and APIs agent so that the Test Integrations and APIs agent can register with Test Hub.



Note: When you configure the Test Integrations and APIs agent by using the offline user token, Test Hub considers you as the owner of the API agent.

- Viewed the API agents that are registered with Test Hub. See [Viewing API agents that are registered with DevOps Test Hub on page 395](#).
- Added the remote API agents that are registered with Test Hub to your project on Test Hub. See [Adding an API agent to a project on page 397](#).

If you do not want to retain the API agents in a project or if you want to add the API agents to a different project, you can remove the API agents from a project. When you remove an API agent from a project, you cannot select that API agent to run virtual services in the project. See [Removing an API agent from a project on page 399](#).

If you want to remove a registered API agent from a team space in Test Hub when the API agent is not available for use, you can unregister a remote API agent from a team space. See [Unregistering a remote API agent from DevOps Test Hub on page 400](#).

Considerations for running virtual services on remote API agents

Before you configure a run of a virtual service resource on a remote API agent from HCL DevOps Test Hub (Test Hub), you must read the considerations that you must take into account.

Supported virtual services

You can run all virtual services that virtualize services on a remote API agent registered with a team space in Test Hub except the following services:

- Virtualized DPL calls in CICS® Transaction Server.
- Virtualized services for IBM® Db2® on IBM® z/OS®.

Prerequisite tasks

Before you run any of the supported virtual services on a remote API agent from a project on Test Hub, you must have completed the following tasks:

- Installed the Test Integrations and APIs Agent on the remote computer. See [Management of API agents on page 394](#).
- Configured the API agent with your offline user token so that the API agent can connect to Test Hub.
- Added the remote API agents to your project on Test Hub. See [Adding an API agent to a project on page 397](#).

Licensing

When a virtual service is run on an API agent, the license server configured on the remote agent computer determine the license server that is used by the API agent. This might be different from the license server that is configured for the team space.

Viewing API agents that are registered with DevOps Test Hub

After you install the remote HCL DevOps Test Integrations and APIs (Test Integrations and APIs) agents and configure them with your offline user token, Test Integrations and APIs agents register with HCL DevOps Test Hub (Test Hub). You can then view the registered API agents on the **Agents and Intercepts** page.

Before you begin

You must have completed the following tasks:

- Installed the Test Integrations and APIs agent. See [Management of API agents on page 394](#).
- Configured the Test Integrations and APIs agent so that the Test Integrations and APIs agent can register with Test Hub. See [Management of API agents on page 394](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane.

The **Agents and Intercepts** page is displayed.

You can view the API agents, Dockers, intercepts, Kubernetes clusters, or performance agents that are registered with Test Hub.

3. You can view the API agents that you own in any of the following ways:

- Search for the API agent by entering the name of the API agent in the **Search** field.



Note: You can enter either the full name or any text that is in the name. The search is enabled for case-sensitive text that you can enter.

- Sort the **Type** column to sort the items and display the API agents in the rows at the top of the table.
- Sort the **Agents** column and identify the API agents by their names or by their owners.

4. Perform the following steps to view the details and capabilities of the API agents:

- a. Expand the API agent to view the details panel.

You can find the following details of the API agent that is displayed:

Parameter	Description
Agent ID	The ID of the installed API agent.
Identifier	The unique identifier of the installed API agent that is configured in the <code>Agent.config</code> file for the API agent.
Hostname	The hostname of the computer on which the API agent is installed.
Version	The version of the installed API agent.
Operating system	The operating system of the computer on which the API agent is installed.
Configuration location	The path to the location of the <code>Agent.config</code> file on the computer where the API agent is installed.
Tags	The agent attributes or tags that exist in the <code>Agent.config</code> file.

- b. Expand the capabilities panel to view the system capabilities.

You can find the following capabilities of the API agent displayed:

Capabilities	Description
System capabilities	The capabilities of the API agent.

- c. Check for the status of the API agent from the **Status** column.

Results

You have viewed the registered API agents from the **Agents and Intercepts** page.

What to do next

You can add the API agents that you own to your project. See [Adding an API agent to a project on page 397](#).

Adding an API agent to a project

You must add the registered API agents to your projects before you select them as the location to run the virtual service resources that are in your project on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Installed the agent by using your offline user token that makes you the owner of the agent or you are assigned the role of a *Team Space Owner*.

About this task

An owner of an API agent or team space owner can add or remove an API agent from the project. However, any member of the project with the permission to run the virtual services can initiate the run of the virtual service resources on API agents that are added to the project.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open your project, and then click **Infrastructure > Agents and Intercepts**.

The **Agents and Intercepts** page is displayed.

3. Click **Add > Add API agent**.

The API agents that you configured with your user token are displayed.



Note: You can add the API agents that you own. You can add any number of API agents to your project.

4. Select the API agents that you want to use in the project, and then click **Add**.

The API agents that you added to the project are displayed.

5. Perform the following steps to view the details and capabilities of the API agents:
 - a. Expand the API agent to view the details panel.

You can find the following details of the API agent that is displayed:

Parameter	Description
Agent ID	The ID of the installed API agent.
Identifier	The unique identifier of the installed API agent that is configured in the <code>Agent.config</code> file for the API agent.
Hostname	The hostname of the computer on which the API agent is installed.
Version	The version of the installed API agent.
Operating system	The operating system of the computer on which the API agent is installed.
Configuration location	The path to the location of the <code>Agent.config</code> file on the computer where the API agent is installed.
Tags	The agent attributes or tags that exist in the <code>Agent.config</code> file.

- b. Expand the capabilities panel to view the system capabilities.

You can find the following capabilities of the API agent displayed:

Capabilities	Description
System capabilities	The capabilities of the API agent.

- c. Check for the status of the API agent from the **Status** column.

Results

You have added remote API agents to your project and viewed the details and capabilities of the API agent that is displayed for each agent that you own.

What to do next

Configure a run of the virtual service resources that are in your project and select an API agent as the location to run the virtual service. See [Configuring a run of a virtual service on page 750](#).

Removing an API agent from a project

You can remove an API agent from your project in HCL DevOps Test Hub (Test Hub) when you no longer need the agent to run virtual services.

Before you begin

You must have added the registered API agent to a project in your team space. See [Adding an API agent to a project on page 397](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open your project, and then click **Infrastructure > Agents and Intercepts**.

The agents that you added to your project are displayed on the **Agents and Intercepts** page.


3. You can view the API agents that you own in any of the following ways:

- Search for the API agent by entering the name of the API agent in the **Search** field.




Note: You can enter either the full name or any text that is in the name. The search is enabled for case-sensitive text that you can enter.

- Sort the **Type** column to sort the items and display the API agents in the rows at the top of the table.
- Sort the **Agents** column and identify the API agents by their names or by their owners.

4. Identify the API agent that you want to remove from the project, and then click the **Remove agent** icon .



Note: The  icon is displayed in the **Actions** column in the row of the API agent only if you are the owner of the remote API agent.

Results

You have removed the registered API agent from your project. The API agent is no longer available as a location to you while you configure a run of virtual service resources that are in your project.

What to do next

You might have to add API agents to your project to use the remote API agents as a location to run tests, if there are no API agents added to your project.

Related information

[Unregistering a remote Docker host from DevOps Test Hub on page 416](#)

Unregistering a remote API agent from DevOps Test Hub

You can unregister a remote API agent that is registered with HCL DevOps Test Hub (Test Hub) in a team space when you no longer require it.

Before you begin

You must have completed the following tasks:

- Installed the agent by using your offline user token that makes you the owner of the agent or you are assigned the role of a *Team Space Owner*.
- Configured the Test Integrations and APIs agent so that the Test Integrations and APIs agent can register with Test Hub. See [Management of API agents on page 394](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.



2. Click **Infrastructure > Agents and Intercepts** in the navigation pane.

The **Agents and Intercepts** page is displayed.


You can view the API agents, Dockers, intercepts, Kubernetes clusters, or performance agents that are registered with Test Hub.

3. Identify the API agent that you want to unregister.



Note: The  icon is displayed in the **Actions** column in the row of the API agent only if you are the owner of the API agent or you are assigned the role of a *Team Space Owner*. The  icon is enabled only if the API agent is in a *Lost Contact* state.

4. Unregister the API agent by completing the following steps:

- a. Click the **Unregister API agent** icon .
- b. Click **Unregister** in the **Unregister API agent** dialog box.

Results

You have unregistered the remote API agent from a team space in Test Hub. You cannot run tests on this remote API agent if it is added to your project.



Note: You might have to register it again with Test Hub, if you want to use the remote API agent in your projects.

Related information

[Removing a remote Docker host from a project on page 414](#)

Management of Docker hosts

You can find information about the tasks that you can perform on remote Docker hosts. After you set up a remote Docker host, you can register the Docker host with HCL DevOps Test Hub (Test Hub). You must then add the registered remote Docker host to your project before you can run tests on the remote Docker host location.

You can find the following information about managing Dockers:

- [Test run considerations for running tests on remote Docker hosts on page 401](#)
- [Setting up a remote Docker host computer on page 403](#)
- [Setting up a secure remote Docker host computer on page 404](#)
- [Copying third-party application Jars to a remote Docker host on page 406](#)
- [Registering a remote Docker host on page 408](#)
- [Viewing remote Docker hosts that are registered with DevOps Test Hub on page 410](#)
- [Adding a remote Docker host to the project for running tests on page 411](#)
- [Editing configurations of a remote Docker host on page 412](#)
- [Removing a remote Docker host from a project on page 414](#)
- [Unregistering a remote Docker host from DevOps Test Hub on page 416](#)

Test run considerations for running tests on remote Docker hosts

Before you can configure a test to run on a remote Docker host, you must read the considerations that you must take into account.

You can run all the tests that are supported on HCL DevOps Test Hub (Test Hub) on remote Docker hosts.

When you plan to run any of the tests on a remote Docker host from a project on Test Hub, you must complete the following tasks:

	Tasks	More information...
1.1	Set up non-secure remote Docker hosts.	Setting up a remote Docker host computer on page 403
1.2	Set up secure remote Docker hosts.	Setting up a secure remote Docker host computer on page 404
2	Register the remote Docker host on Test Hub.	Registering a remote Docker host on page 408

	Tasks	More information...
3	View the registered Docker hosts.	Viewing remote Docker hosts that are registered with DevOps Test Hub on page 410
4	Add a remote Docker host to the Test Hub project.	Adding a remote Docker host to the project for running tests on page 411
5	<p>Configure a run for your test that you want to run on the remote Docker host by performing the following actions:</p> <ol style="list-style-type: none"> 1. Go to Test run configurations on page 446. 2. Check for the prerequisite tasks that you must perform before you configure a test run for the test. 3. Select the task for the test that you want to configure for a run, and then follow the steps to run the test. 	<ol style="list-style-type: none"> 1. Test run configurations on page 446 2. Prerequisites to running tests on page 371 3. Test run configurations on page 446

If you plan to run API Suites that use a transport and the transport requires third-party application `Jar` files for a successful run on a remote Docker host, then you must perform the following task:

- You must copy the third-party application `Jar` files to the third-party application folder on the computer where the remote Docker host is installed. See [Copying third-party application Jars to a remote Docker host on page 406](#).

After you set up and register the remote Docker host with Test Hub, you can add the remote Docker hosts to your project on Test Hub. See [Adding a remote Docker host to the project for running tests on page 411](#).

You can find information about the tasks that you can perform when you use remote Docker hosts that are set up and registered with Test Hub.

- When you configure a test run and in the step in which you select the remote Docker, if you encounter any of the following conditions because these were changed on the remote Docker host:
 - The hostname or port of the remote Docker is not displayed correctly.
 - You want to change the mode of authentication with the remote Docker host.

You must edit the configuration settings for the remote Docker host. See [Editing configurations of a remote Docker host on page 412](#).

- When you want to delete a remote Docker host from your Test Hub project, see [Removing a remote Docker host from a project on page 414](#).
- When you want to remove or unregister a remote Docker host from Test Hub, see [Unregistering a remote Docker host from DevOps Test Hub on page 416](#).

Related information

[Test run configurations on page 446](#)

[Management of Docker hosts on page 401](#)

Setting up a remote Docker host computer

You must set up a computer to host the remote Docker host. You can set up the remote host with a non-secure mode of connection with HCL DevOps Test Hub (Test Hub).

Before you begin

You must have identified a remote computer or VM that has a minimum of 16 GB RAM. You must have installed Ubuntu V18.04 or later, on the remote computer.

You must have installed Docker Enterprise Edition for a non-secure mode of connection with Test Hub on the remote Docker host computer. Refer to [Get Docker Engine](#).

You must have the entitlement key that is required for you to connect to the Harbor repository to be able to pull in the software image to the remote Docker host.

About this task

You must set up a remote computer or a Virtual Machine (VM) on which you want to set up the remote Docker host. You must be a system administrator or get the remote Docker host set up by a system administrator. You must ensure that the system administrator completes the following actions:

- Enables SSH.
- Creates user credentials for you with root permissions.
- Provides the name of the Docker container.
- Provides the IP address of the remote Docker host computer.



Tip: You can issue a ping command by using the hostname or IP address of the remote Docker host computer to verify that the remote host is running and accessible.

1. As a system administrator of the remote system, specify a port to access the Docker daemon in the `docker.service` file located in the directory `/lib/systemd/system`. You must add the configured port at the line in the file:

```
ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:PORT
```

For example, if the port configured is 4342, then the line must read as:

```
ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:4342
```

2. Restart the Docker service by using the following commands:

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl restart docker
```

3. Perform the following steps as a user with root permissions or ask the system administrator to complete the following steps:

a. Open a terminal and connect to the remote Docker host by using an SSH tool by running the following command:

```
ssh root@<IP_address_of_remote_host>
```

b. Log in to HCLSoftware's Harbor container registry by running the following command:

```
sudo docker login hclcr.io -u <username> -p <password>
```



Note: You must provide the following username and password in the command to log in to the registry:

- Username as email address registered with HCL
- Password as Harbor CLI secret

c. Run the following commands to pull the product images from the registry on to the remote Docker host computer:



Note: You must pull in the images with the version which is the same as the version of Test Hub that you use to run the tests.

- `sudo docker pull cp.hclcr.io/ot/hcl-onetest/studio<image_identifier>`
- `sudo docker pull hclcr.io/ot/hcl-onetest/virtualization<image_identifier>`

Results

You have set up the remote Docker host computer that can be accessed in a non-secure mode.

What to do next

You can set up a secure remote Docker host computer optionally, that enables a secure connection with Test Hub. See [Setting up a secure remote Docker host computer on page 404](#).

You can register the remote Docker host on Test Hub. See [Registering a remote Docker host on page 408](#).

Setting up a secure remote Docker host computer

You must set up a computer to host the remote Docker host. You can set up the remote host with a secure mode of connection with HCL DevOps Test Hub (Test Hub) by using certificates issued by the remote host computer that authenticate the connection.

Before you begin

You must have identified a remote computer or VM that has a minimum of 16 GB RAM. You must have installed Ubuntu V18.04 or later, on the remote computer.

You must have installed Docker Community Edition V18.09 or later. For instructions, refer to [Get Docker Engine](#).



Note: For the remote Docker host to be reached in a safe manner through the network, you can enable *TLS* by specifying the **tlsverify** flag and by pointing the **tlscacert** flag of the remote Docker host to a trusted CA certificate. Refer to [Protect the Docker daemon socket](#).

You must have the entitlement key that is required for you to connect to the Harbor repository to be able to pull in the software image to the remote Docker host.

About this task

You must set up a remote computer or a Virtual Machine (VM) on which you want to set up the remote Docker host. You must be a system administrator or get the remote Docker host set up by a system administrator. You must ensure that the system administrator completes the following actions:

- Enables SSH.
- Creates user credentials for you with root permissions.
- Provides the name of the Docker container.
- Provides the IP address of the remote Docker host computer.



Tip: You can issue a ping command by using the hostname or IP address of the remote Docker host computer to verify that the remote host is running and accessible. You can also verify if the default port for the remote Docker host is 2376 by checking that `dockerd` uses port 2376 on the remote computer.

Perform the following steps as a user with root permissions or ask the system administrator to complete the following steps:

- a. Open a terminal and connect to the remote Docker host by using an SSH tool by running the following command:

```
ssh root@<IP_address_of_remote_host>
```

- b. Log in to HCLSoftware's Harbor container registry by running the following command:

```
sudo docker login hclcr.io -u <username> -p <password>
```



Note: You must provide the following username and password in the command to log in to the registry:



- Username as email address registered with HCL
- Password as Harbor CLI secret

c. Run the following commands to pull the product images from the registry on to the remote Docker host computer:



Note: You must pull in the images with the version which is the same as the version of Test Hub that you use to run the tests.

- `sudo docker pull cp.hclcr.io/ot/hcl-onetest/studio<image_identifier>`
- `sudo docker pull hclcr.io/ot/hcl-onetest/virtualization<image_identifier>`

Results

You have set up the remote Docker host computer that can be accessed in a secure mode.

What to do next

You can optionally, set up a non-secure remote Docker host. See [Setting up a remote Docker host computer on page 403](#).

You can register the remote Docker host on Test Hub. See [Registering a remote Docker host on page 408](#).

Copying third-party application Jars to a remote Docker host

You can run API Suites in a project on HCL DevOps Test Hub (Test Hub) on a remote Docker host. If the API Suites use a transport and the transport requires third-party `Jar` files for a successful run, you must ensure that the third-party application `Jar` files are available at the test run time. To achieve this, you must copy the third-party application `Jar` files to the computer where you have set up the remote Docker host.

Before you begin

If you want to copy the third-party application `Jar` files to the to the computer where Test Hub is installed on Kubernetes, see [Copying third-party application Jars to Kubernetes on page 160](#).

You must have server administrator privileges.

You must have completed the following tasks:

- Set up and configured the remote Docker host. See [Management of Docker hosts on page 401](#).
- Identified the third-party application `Jar` files that are required and copied the `Jar` files. See [Test run considerations for API Suites or API tests on page 373](#).
- Copied the `Jar` files of the third-party application jars to the directory or folder on the remote Docker host from where you can run the docker commands.

About this task

You can copy the required third-party application `jar` files to the folder that is specific for the application under the `/myFiles/Userlibs/<application_name>` folder. You need not extract the files.

You can perform this task any time after you have configured the remote Docker host and you plan to run an API Suite on the remote Docker host. The API Suite uses transports and the transports require third-party application `JAR` files to be available at the test run time.

You can get help on the docker commands by running the command: `$ docker help` from the docker command line. For more information about the docker commands, refer to the [Docker command line documentation](#).

1. Use the following table to find the name of the folder that corresponds to the specific third-party application for the transport used in the API Suite.



Note: You must provide the name of the folder listed for the third-party application as the `<application_name>` in the docker command.

Table 21. Name of the folder for the application

Application	Name of the folder to use
Camel	Camel
CentraSite	CentraSite
CICS	CICS
Coherence	Coherence
Database	JDBC
IMS	IMS
Integra	Integra
JMS	JMS
SAP RFC	SAP
Software AG Universal Messaging	SoftwareAGUM
TIBCO EMS	TIBCO
TIBCO Rendezvous	
TIBCO SmartSockets	
WebSphere Application Server Service Integration Bus (SiBus)	WAS
WebLogic	WebLogicJMX
Software AG webMethods	webMethods

2. Run the following docker command to create or update the volume that contains the application `Jar` files:

```
docker run --rm -v /<anyFolder>/UserLibs/<application_name>:/ulsrc -v userlibs/<application_name>:/uldest  
alpine:latest cp -r /ulsrc/. /uldest/UserLibs/<application_name>
```



Attention: You must run the docker command when no other tests are running in the Docker host to prevent concurrent access problems.

Results

You have successfully copied the third-party application `Jar` files to the folder in the remote Docker host.

What to do next

- You can register the remote Docker host, if you have not already done so. See [Registering a remote Docker host on page 408](#).
- You can configure the API Suite run. See [Configuring an API Suite or an API test run on page 486](#).

Registering a remote Docker host

You must register the remote Docker host with HCL DevOps Test Hub (Test Hub) before you add the remote Docker host to your project in a team space, which contains the test assets that you want to run on the remote Docker host.

Before you begin

Depending on the mode of connection with Test Hub, you must have set up the remote Docker host computer either for a secured mode or for an unsecured mode of connection.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane.

The **Agents and Intercepts** page is displayed.

You can view the agents, intercepts or Dockers that are registered with Test Hub.

3. Click **Add new > Add new docker host**.

The **New docker host** dialog is displayed.

4. Enter the *hostname* and the *port* of the remote Docker host in the format `<hostname or IP_address>:<port>` in the **Remote Docker Host** field.

If you do not provide the port that you configured for the remote Docker host, then the default port that is used is as follows:

- Port 2375 for the unsecured mode
- Port 2376 for the secured mode

5. Choose from any of the following authentication modes to establish the connection between Test Hub and the remote Docker host computer:

- For the secured mode, which is the default mode and the checkbox is displayed as selected.

Perform the following steps:

- a. Select **Secure mode**, if it is not selected.
- b. Click the action labels to browse and select the `.pem` files that correspond to `ca_certificate.pem`, `client_certificate.pem`, and `client_key.pem`.



Note: You must have generated the `.pem` files on the remote Docker host computer and copied the files to your local drive.

- For the unsecured mode, the **Secure mode** checkbox must not be selected.

Perform the following step:

- a. Clear the **Secure mode** checkbox, if selected.

6. Click **Test connection** to test if a connection is established between Test Hub and the remote host computer.

Any of the following events occur when **Test connection** is clicked:

- On a successful connection, a message is displayed.
- On a failure to connect to the remote host computer, an error message is displayed. You must resolve the error and reattempt to establish a successful connection. You can troubleshoot the connection errors by attempting the solutions provided in the troubleshooting topic.



Important: The remote Docker host computer must be connected successfully before you register the remote Docker host with Test Hub.

7. Click **Register**.

The remote Docker host that you registered is displayed.



Important: The remote Docker host must be registered successfully with Test Hub in the team space before you add the remote Docker host to your project in that team space.

8. Click **Close** to exit.

Results

You have registered the remote Docker host in your team space with Test Hub.

What to do next

You can view the Docker hosts in your team space that are registered with Test Hub. See [Viewing remote Docker hosts that are registered with DevOps Test Hub on page 410](#).

You must add the registered remote Docker host to your project in your team space before you use the remote Docker host as a location to run tests in your project. See [Adding a remote Docker host to the project for running tests on page 411](#).

Related information

[Table 3 on page 1080](#)

Viewing remote Docker hosts that are registered with DevOps Test Hub

You must register the remote Docker host in your team space with HCL DevOps Test Hub (Test Hub) before you add the remote Docker host to your project in that team space, which contains the test assets that you want to run on the remote Docker host.

Before you begin

Depending on the mode of connection with Test Hub, you must have set up the remote Docker host computer either for a secured mode or for an unsecured mode of connection. See [Setting up a remote Docker host computer on page 403](#) or [Setting up a secure remote Docker host computer on page 404](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane.

The **Agents and Intercepts** page is displayed.

You can view the agents, intercepts or Dockers that are registered with Test Hub in your team space.

3. You can view the Dockers that you own in any of the following ways:

- Search for the Docker by entering the name of the Docker host in the **Search** field.



Note: You can enter either the full name or any text that is in the name. The search is enabled for case sensitive text that you can enter.

- Sort the **Type** column to sort the items and then identify the Docker by the name displayed.
- Sort the **Agents** column to sort the items and identify the Docker by the owner.

You can view the following details about the Dockers that are registered with Test Hub in the team space:

- The projects in the team space to which the Docker host is added, are displayed in the **Projects** column.
- The status of the Docker host is displayed in the **Status** column.



Note: You can add Docker hosts that are in the `Ready` state to your project, and then use them as locations to run tests.

Results

You have viewed the remote Docker hosts that are registered with Test Hub in your team space.

What to do next

You must add the registered remote Docker host to your project in your team space before you use the remote Docker host as a location to run tests in your project. See [Adding a remote Docker host to the project for running tests on page 411](#).

Adding a remote Docker host to the project for running tests

You can choose to run tests on Docker hosts that you have set up on remote host computers. You must register the Docker host with HCL DevOps Test Hub (Test Hub) and then add them to your project before you run tests on the remote Docker hosts.

Before you begin

You must have completed the following tasks:

- Read about the considerations you must take into account before you configure a test run to run on a remote Docker host. See [Test run considerations for running tests on remote Docker hosts on page 401](#).
- Set up the remote Docker host system. See [Setting up a remote Docker host computer on page 403](#) or [Setting up a secure remote Docker host computer on page 404](#).
- Registered the remote Docker host with Test Hub. See [Registering a remote Docker host on page 408](#).

1. Log in to Test Hub.
2. Open your project.
The **Overview** page is displayed.
3. Click **Infrastructure > Agents and Intercepts**.

The **Agents and Intercepts** page is displayed.

4. Click **Add > Add docker**.
5. Select the Docker host that you want to add to the project from the list of Docker hosts, and then click **Add**.




Note: You can add the Docker hosts that you registered. You can add any number of Docker hosts to your project.

The Docker hosts that you added to the project are displayed.

You can view the details of the Docker host by clicking the **Expand** icon

The screenshot shows the 'Agents' page for a project named 'proj1'. At the top, there is a search bar with the text 'Search'. Below it is a table with the following structure:

Type	Agents																								
	<p>10.121.104.38:80 Owned by me</p> <p>Details</p> <table border="0"> <tr><td>Server:</td><td>Docker Engine - Community</td></tr> <tr><td>Secure</td><td>No</td></tr> <tr><td>Version:</td><td>19.03.2</td></tr> <tr><td>Address:</td><td>10.121.104.38</td></tr> <tr><td>Port:</td><td>80</td></tr> <tr><td>Operating system:</td><td>linux</td></tr> <tr><td>Architecture:</td><td>amd64</td></tr> <tr><td>Api version:</td><td>1.40 (minimum version 1.12)</td></tr> <tr><td>Go version:</td><td>go1.12.8</td></tr> <tr><td>Git commit:</td><td>6a30dfc</td></tr> <tr><td>Built:</td><td>Thursday, August 29, 2019 5:27 AM</td></tr> <tr><td>Experimental:</td><td>false</td></tr> </table>	Server:	Docker Engine - Community	Secure	No	Version:	19.03.2	Address:	10.121.104.38	Port:	80	Operating system:	linux	Architecture:	amd64	Api version:	1.40 (minimum version 1.12)	Go version:	go1.12.8	Git commit:	6a30dfc	Built:	Thursday, August 29, 2019 5:27 AM	Experimental:	false
Server:	Docker Engine - Community																								
Secure	No																								
Version:	19.03.2																								
Address:	10.121.104.38																								
Port:	80																								
Operating system:	linux																								
Architecture:	amd64																								
Api version:	1.40 (minimum version 1.12)																								
Go version:	go1.12.8																								
Git commit:	6a30dfc																								
Built:	Thursday, August 29, 2019 5:27 AM																								
Experimental:	false																								

What to do next

You can select any of the remote Docker hosts as an alternate location to run the test asset in your project while configuring a test run from the **Execution** page.

Related information

[Test run configurations on page 446](#)

Editing configurations of a remote Docker host

You can edit the hostname or the port of the registered Docker host computer instead of registering it as a new remote host if any of the parameters are changed on the remote host computer. You can also change the mode of authentication of a registered remote Docker host.

Before you begin

You must have completed the following tasks:

- Set up the remote Docker host computer. See [Setting up a remote Docker host computer on page 403](#) or [Setting up a secure remote Docker host computer on page 404](#).
- Registered the remote Docker host with HCL DevOps Test Hub (Test Hub) in your team space. See [Registering a remote Docker host on page 408](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane.

The **Agents and Intercepts** page is displayed.

You can view the agents, intercepts, Dockers, or clusters that are registered with Test Hub in your team space.

3. You can view the Dockers that you own in any of the following ways:

- Search for the Docker by entering the name of the Docker host in the **Search** field.




Note: You can enter either the full name or any text that is in the name. The search is enabled for case sensitive text that you can enter.


- Sort the **Type** column to sort the items and then identify the Docker by the name displayed.
- Sort the **Agents** column to sort the items and identify the Docker by the owner.

You can view the following details about the Dockers that are registered with Test Hub in your team space:

- The projects in the team space to which the Docker host is added, are displayed in the **Projects** column.
- The status of the Docker host is displayed in the **Status** column.

4. Identify the Docker host that you want to modify the configuration, and then click the **Edit** icon .




Note: The  icon is displayed in the **Actions** column in the row of the Docker only if you own the remote Docker host.

The **Update docker host** dialog is displayed.

5. Change any of the configurations.

To change...	Do this...
Change the hostname or the port of the remote Docker host.	Enter the changed value of the hostname or the port in the Remote hostname field.
Change the mode of authentication from a non-secure mode to a secure mode.	a. Select the Secure mode option. b. Click the action labels to browse and select the <code>.pem</code> files that correspond to <code>ca_</code>

To change...	Do this...
	<p>certificate.pem, client_certificate.pem, and client_key.pem.</p> <p> Note: You must have generated the .pem files on the remote host computer and copied the files to your local drive.</p>
Change the mode of authentication from a secure mode to a non-secure mode.	Clear the Secure mode option that is selected.

6. Click **Test connection** to test whether a connection is established between Test Hub and the remote host computer:

- On a successful connection, a message is displayed.
- On a failure to connect to the remote host computer, an error message is displayed. You must resolve the error and reattempt to establish a successful connection.

7. Click **Update**.

The remote Docker host with the updated details is displayed.

Results

You have updated the registered remote Docker host with the changed parameters.

What to do next

You can now add the updated Docker host to your project. See [Adding a remote Docker host to the project for running tests on page 411](#).

Related information

[Setting up a remote Docker host computer on page 403](#)

[Registering a remote Docker host on page 408](#)

[Removing a remote Docker host from a project on page 414](#)

[Unregistering a remote Docker host from DevOps Test Hub on page 416](#)

Removing a remote Docker host from a project

You can delete a remote Docker host that you added to your project in HCL DevOps Test Hub (Test Hub) when you no longer need it to run tests.

Before you begin

You must have completed the following tasks:

1. Set up the remote Docker host computer. See [Setting up a remote Docker host computer on page 403](#) or [Setting up a secure remote Docker host computer on page 404](#).
2. Registered the remote Docker host with Test Hub in your team space. See [Registering a remote Docker host on page 408](#).
3. Added the registered Docker host to a project in your team space. See [Adding a remote Docker host to the project for running tests on page 411](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project in your team space from the **My Projects** page.

The **Overview** page is displayed.

3. Click **Infrastructure > Agents and Intercepts** in the navigation pane.

The **Agents and Intercepts** page is displayed.

You can view the agents, intercepts or Dockers that are added to your project.

4. You can view the Dockers that you own in any of the following ways:
 - Search for the Docker by entering the name of the Docker host in the **Search** field.




Note: You can enter either the full name or any text that is in the name. The search is enabled for case sensitive text that you can enter.


- Sort the **Type** column to sort the items and then identify the Docker by the name displayed.
- Sort the **Agents** column to sort the items and identify the Docker by the owner.

You can view the following details about the Dockers that are registered with Test Hub:

- The projects to which the Docker host is added, are displayed in the **Projects** column.
- The status of the Docker host is displayed in the **Status** column.

5. Identify the Docker host that you want to remove from the project, and then click the **Remove docker host** icon .



Note: The  icon is displayed in the **Actions** column in the row of the Docker only if you own the remote Docker host.

A message is displayed that the Docker host is removed from the project successfully.

Results

You have removed the remote Docker host from your project. The remote Docker host is no longer available as a location while configuring a test run in your project.

What to do next

You might have to add a remote Docker host to your project to use the remote Docker host as a location to run tests.

Related information

[Unregistering a remote Docker host from DevOps Test Hub on page 416](#)

[Editing configurations of a remote Docker host on page 412](#)

Unregistering a remote Docker host from DevOps Test Hub

You can unregister a remote Docker host that is registered with HCL DevOps Test Hub (Test Hub) in a team space when you no longer require it.

Before you begin

You must have completed the following tasks:

1. Set up the remote Docker host computer. See [Setting up a remote Docker host computer on page 403](#) or [Setting up a secure remote Docker host computer on page 404](#).
2. Registered the remote Docker host with Test Hub in a team space. See [Registering a remote Docker host on page 408](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.



2. Click **Infrastructure > Agents and Intercepts** in the navigation pane.

The **Agents and Intercepts** page is displayed.


You can view the agents, intercepts, Dockers, or clusters that are registered with Test Hub in your team space.

3. Identify the Docker host that you want to unregister.



Note: The  and  icons are displayed in the **Actions** column in the row of the Docker only if you own the remote Docker host.

4. Unregister the Docker host by completing the following steps:

- a. Click the **Unregister docker host** icon .
- b. Click **Unregister** in the **Unregister docker host** dialog box.

A message is displayed that the Docker host is unregistered successfully.

Results

You have unregistered the remote Docker host in a team space on Test Hub. You cannot run tests on this remote Docker host if it is added to your project. You might have to register it again with Test Hub, if you want to use the remote Docker host in your projects.

Related information

[Editing configurations of a remote Docker host on page 412](#)

[Adding a remote Docker host to the project for running tests on page 411](#)

[Removing a remote Docker host from a project on page 414](#)

Management of remote Kubernetes clusters

You can find information about the tasks that you can perform on remote Kubernetes clusters. After you set up a remote Kubernetes cluster, you can register the external cluster with a team space on HCL DevOps Test Hub (Test Hub). You must then add the remote cluster to a project before you run test assets or resources on the remote Kubernetes cluster.

You can find the following information about managing the external Kubernetes cluster:

- [Considerations for using a remote Kubernetes cluster on page 417](#)
- [Registering a remote Kubernetes cluster on page 420](#)
- [Viewing remote Kubernetes clusters on page 423](#)
- [Adding a remote Kubernetes cluster to a project on page 425](#)
- [Editing configurations of a remote Kubernetes cluster on page 427](#)
- [Removing a remote Kubernetes cluster from a project on page 429](#)
- [Unregistering a remote Kubernetes cluster from a team space on page 430](#)

Considerations for using a remote Kubernetes cluster

When you want to use a remote Kubernetes cluster as a location to run the supported tests or test resources, you must read the considerations first.

Before you want to run any of the supported tests or test resources on HCL DevOps Test Hub (Test Hub) that are configured to run on a remote Kubernetes cluster, you must have completed the following tasks:

- Verified that the Kubernetes cluster that you want to use is a supported one by referring to the OS, platform, and containers that are specified for Test Hub in [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).
- Kept a suitable `kubeconfig` file with access to the remote Kubernetes cluster in a folder on the computer that you want to use to access an instance of Test Hub.

You must be familiar with the administration of a Kubernetes cluster. When you as the cluster administrator, set up the remote Kubernetes cluster must consider the following requirements:

- You must have a `kubeconfig` file with access to the remote Kubernetes cluster in a folder on the computer that you want to use to access an instance of Test Hub.



Note: If the `kubeconfig` file references certificate files by using filepaths then the certificate data must be embedded in the `kubeconfig` file to make it usable by Test Hub. If the `kubeconfig` file contains multiple contexts, then only the current context is required.

You can use the following `kubectl` command to create a suitable `kubeconfig` file called `portable-config` in the current folder. The `portable-config` file contains embedded certificate data and the information used by the current context:

```
kubectl config view --flatten --minify > portable-config
```

- You must ensure that the remote Kubernetes cluster is able to access the Test Hub image registry which is the same registry that you used to install Test Hub. There is no need to copy the Test Hub image to the remote Kubernetes cluster but the remote cluster must be able to access the image registry at test runtime.

If the remote cluster that you want to use is a Lightweight Kubernetes (k3s) cluster, it might require a `Secret` to pull the Test Hub image. You must run the following command on the remote k3s cluster:

```
kubectl create secret docker-registry hclcr.io \
-n {remote-namespace} \
--docker-server=hclcr.io \
--docker-username={okta-email-address} \
--docker-password={harbor-cli-secret} \
--docker-email={example@abc.com}
```



Notes:

- You must replace `{remote-namespace}` with the value of the `namespace` that you have used for the remote cluster.
 - You must replace `{okta-email-address}` with the username of the Harbor repository and replace `{harbor-cli-secret}` with the secret key that you copied from the Harbor repository.
 - If the username contains any special characters, such as `$`, you must enclose it within single quotation-marks.
 - You can replace `{example@abc.com}` with the email address of the administrator, if required.
- You need not configure separate licenses for running on the remote cluster because the licenses are consumed from the pool of available licenses that are configured for the team space on Test Hub. The licenses are returned to the pool when the test run is completed on the remote cluster.
 - You must ensure that the Test Hub domain is accessible from the remote cluster so that the pod in the cluster is able to communicate with Test Hub.
 - You must ensure that the appropriate **PriorityClass** exists on the remote cluster if you want to control the priority given to the remote execution pod by the scheduler. The same **PriorityClass** that is used by the Test

Hub execution pod is used by the remote execution pod. If the named **PriorityClass** does not exist in the remote cluster, then no **PriorityClass** is specified by the remote execution pod.

- You must create an HTTP or an HTTPS Emissary listener when you want to run virtual services on the remote cluster. You must create both listeners when you want to run virtual services that use both HTTP and HTTPS transports. For information about creating listeners, refer to <https://www.getambassador.io/docs/emissary/latest/topics/running/listener>. The Emissary listener associates with the hosts created by Test Hub in the target namespace for executions that require Ingress. For example, when you want to run virtual services on the remote cluster and run tests from the remote cluster that use static agents.
- You can enter the code snippets for *nodeSelectors* and *tolerations* in a `yaml` format, if you want to use *nodeSelectors* and *tolerations* on the remote cluster. You can enter these in the fields provided in the **Advanced** configuration dialog when you register a remote cluster with your team space on Test Hub.

For more information about *nodeSelector*, see [nodeSelector](#) and for *tolerations*, see [Tolerations](#).



Restrictions: The following restrictions apply when you want to run the supported tests on a remote Kubernetes cluster:

- Using the **UserLibs** that are required for running certain API Suites are not supported on the remote cluster.
- Running of Compound Tests that contain performance tests, which require the **NodePort** access to connect to static agents, are not supported on the remote cluster.

When you plan to run any of the test assets or resources on a remote Kubernetes cluster from a project on Test Hub, you must complete the following tasks:

	Tasks	More information...
1	Register a remote Kubernetes cluster with a team space on Test Hub.	Registering a remote Kubernetes cluster on page 420
2	View remote Kubernetes clusters added to a team space.	Viewing remote Kubernetes clusters on page 423
3	Add a remote Kubernetes cluster to a project in a team space.	Adding a remote Kubernetes cluster to a project on page 425
4.1	Configure a run for your test that you want to run on the remote Kubernetes cluster by performing the following actions:	<ol style="list-style-type: none"> 1. Test run configurations on page 446 2. Prerequisites to running tests on page 371 3. Test run configurations on page 446

	Tasks	More information...
	<ol style="list-style-type: none"> 1. Check for the prerequisite tasks that you must perform before you configure a test run for the test. 2. Select the task for the test that you want to configure for a run, and then follow the steps to run the test. 	
4.2	<p>Configure a run of the virtual service resources that you want to run on the remote Kubernetes cluster by performing the following actions:</p> <ol style="list-style-type: none"> 1. Check for the prerequisite tasks that you must perform before you configure a run of the virtual services. 2. Select the virtual service resources in your project that you want to configure for a run, and then follow the steps to run them. 	<ol style="list-style-type: none"> 1. Prerequisites for running HTTP virtual services on page 742 2. Configuring a run of a virtual service on page 750

Related information

[Test run configurations on page 446](#)

[Management of remote Kubernetes clusters on page 417](#)

Registering a remote Kubernetes cluster

You must register a remote Kubernetes cluster to your team space on HCL DevOps Test Hub (Test Hub) before you add the remote cluster to your project.

Before you begin

You must have completed the following tasks:

- Verified that the Kubernetes cluster that you want to use is a supported one by referring to the OS, platform, and containers that are specified for Test Hub in [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).
- Kept a suitable `kubeconfig` file with access to the remote Kubernetes cluster in a folder on the computer that you want to use to access an instance of Test Hub.

- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane of the team space.

The **Agents and Intercepts** page is displayed.

3. Click **Add new > Add new cluster**.

The **New cluster** dialog is displayed.

4. Perform the following steps to add the `kubeconfig` file:

a. Click **Browse**, and locate the `kubeconfig` file of the Kubernetes cluster that you want to add.

b. Select the `kubeconfig` file, and then click **Open**.

The name of the `kubeconfig` file that you selected is displayed along with the address of the Kubernetes cluster that is specified in the file.

c. Click **Next**.

The address of the Kubernetes cluster that is configured in the `kubeconfig` file is displayed in the **Address** field.



Note: You can change the address of the Kubernetes cluster if it is different from the address in the `kubeconfig` file.

5. Perform the following steps to add a namespace and name for the cluster:

a. Enter the namespace of the cluster in the **Namespace** field.

b. Enter a name for the cluster in the **Name** field.

The name helps you to identify the cluster.

c. Click **Next**.

6. Perform the following steps to add the ingress details of the cluster:

a. Select the ingress type of the cluster from the **Ingress type** list.

You can select the ingress type from the following options that are displayed:

Option	Description	Result
None	When selected specifies that no ingress type is configured for the remote Kubernetes cluster.	You can run any of the supported tests on the remote cluster but not on the remote agents.
NGINX	When selected specifies that NGINX is configured as the ingress type for the remote Kubernetes cluster.	You can run any of the supported tests on the remote cluster or on any of the remote agents.

- b. Enter the ingress host pattern of the cluster in the **Ingress host pattern** field.

The DNS used by a computer that runs a client application (such as a static agent) or the HTTP proxy that routes traffic to the virtual service needs to resolve such hostnames to the IP address of the ingress domain.

For example, if the IP of the remote cluster is 12.12.12.123, then you must enter the ingress host pattern as `*.12.12.12.123.nip.io`

The *Unique ID* of the virtual service is prefixed to the ingress hostname when virtual services are routed from the Test Hub cluster.

- c. Enter the name of the secret in the remote namespace, which contains the certificate information.

You must have created the secret and named it following the naming conventions per the Kubernetes guidelines. The certificate that you create is a wildcard certificate which contains the Ingress host pattern of the Kubernetes cluster. This wildcard certificate ensures that the client applications can trust the virtual services.



Note: If a secret name is not provided then the default certificate that is provided by NGINX is used, which might cause issues to authenticate the virtual service by the client applications.

- d. Click **Next**.


7. Optionally, perform the following steps to enter the code snippets for *nodeSelectors* and *tolerations*, if applicable for the remote cluster:

- Enter the code snippet for *nodeSelectors* in a `yaml` format in the **Node selectors** field.
- Enter the code snippet for *tolerations* in a `yaml` format in the **Tolerations** field.
- Select the **Copy pull secret to the cluster** checkbox, if it is not already selected.



Note: The remote Kubernetes cluster pulls the secret to access the Test Hub image registry which is the same registry that you used to install Test Hub. The remote Kubernetes cluster must be able to access the image registry at test runtime.

d. Click **Register**.

The remote cluster is added to the team space and is displayed with the  icon in the **Type** column on the **Agents and Intercepts** page. You can identify the cluster with its name in the **Agents** column.



Note: The **Projects** column does not display any projects that are associated with the cluster yet.

Results

You have registered a remote Kubernetes cluster with your team space on Test Hub.

What to do next

You must add the remote cluster to your project in your team space before you use the remote cluster as a location to run test assets or resources in your project. See [Adding a remote Kubernetes cluster to a project on page 425](#).

Related information

[Management of remote Kubernetes clusters on page 417](#)

Viewing remote Kubernetes clusters

You can view the clusters that you have registered with a team space on HCL DevOps Test Hub (Test Hub) to view their details or edit their configurations.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Registered a remote Kubernetes cluster with your team space on Test Hub. See [Registering a remote Kubernetes cluster on page 420](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane of the team space.

The **Agents and Intercepts** page is displayed.

You can view the agents, intercepts, Dockers, or clusters that are registered with the team space on Test Hub.

3. You can view the clusters that you own in any of the following ways:

- Search for the cluster by entering the name of the cluster in the **Search** field.



Note: You can enter either the full name or any text that is in the name. The search is enabled for case-sensitive text that you can enter.


- Sort the **Type** column to sort the items, and then identify the cluster by the name or owner.
- Sort the **Agents** column to sort the items, and then identify the cluster by the name or owner.

You can view the following details about the clusters that are in the team space:

- The projects in the team space to which the cluster is added are displayed in the **Projects** column.
- The status of the cluster is displayed in the **Status** column.



Note: You can add clusters that are in the *Ready* state to your project, and then use them as locations to run tests.

You can view the details of the cluster by clicking the **Expand** icon .

The screenshot shows a user interface for viewing cluster details. At the top, there are tabs for 'Type' and 'Agents'. Below this, a cluster named 'mycluster' is shown, owned by 'me'. A 'Details' panel is open, displaying a table with the following information:

Name	Value
Address	https://10.134.54.52:6443
Namespace	default
Nodes	1
Version	1.21

The following table describes the details about the cluster:

Name	Description
Address	Specifies the address of the remote Kubernetes API.

Name	Description
Namespace	Specifies the <i>namespace</i> of the remote Kubernetes cluster in which the test runs occur.
Nodes	Specifies the number of nodes in the remote Kubernetes cluster.
Version	Specifies the Kubernetes version of the remote Kubernetes cluster.

Results

You have viewed details of the remote clusters that are registered with a team space on Test Hub.

What to do next

You must add the remote cluster to your project in your team space before you can use the remote cluster as a location to run test assets or resources that are in your project. See [Adding a remote Kubernetes cluster to a project on page 425](#).

Related information

[Management of remote Kubernetes clusters on page 417](#)

Adding a remote Kubernetes cluster to a project

You must add a remote Kubernetes cluster to your project after you register the cluster with your team space on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Registered a remote Kubernetes cluster with your team space on Test Hub. See [Registering a remote Kubernetes cluster on page 420](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Been assigned a role as an *Owner* in the project.
- Created a project.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open your project.

The **Overview** page is displayed.

- Click **Infrastructure > Agents and Intercepts** in the navigation pane of the team space.

The **Agents and Intercepts** page is displayed.

- Click **Add > Add cluster**.


The **Add cluster** dialog is displayed.

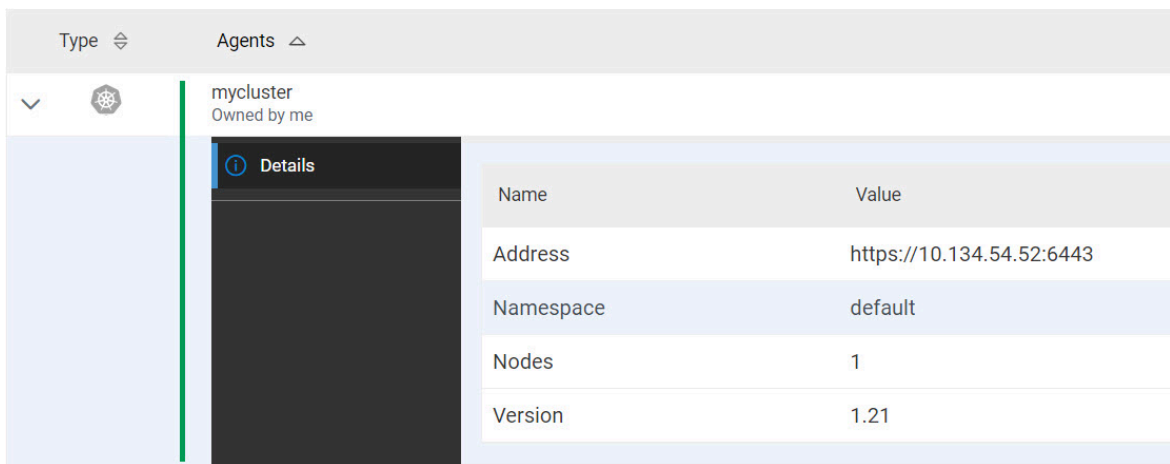
- Select the cluster that you want to add to the project from the list of clusters that are registered with the team space, and then click **Add**.



Note: You can add the clusters that are registered with the team space. You can add any number of clusters to your project.

The clusters that you added to the project are displayed.

You can view the details of the cluster by clicking the **Expand** icon .



Name	Value
Address	https://10.134.54.52:6443
Namespace	default
Nodes	1
Version	1.21

The following table describes the details about the cluster:

Name	Description
Address	Specifies the address of the remote Kubernetes API.
Namespace	Specifies the <i>namespace</i> of the remote Kubernetes cluster in which the test runs occur.
Nodes	Specifies the number of nodes in the remote Kubernetes cluster.
Version	Specifies the Kubernetes version of the remote Kubernetes cluster.

What to do next

You can select any of the remote clusters as an alternate location to run the test assets or resources that are in your project while you configure a run from the **Execution** page.

Related information

[Test run configurations on page 446](#)

[Management of remote Kubernetes clusters on page 417](#)

Editing configurations of a remote Kubernetes cluster

You can edit the `kubeconfig` file, address, namespace, name, or ingress details of the remote Kubernetes cluster that you added to a team space on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Registered a remote Kubernetes cluster with your team space on Test Hub. See [Registering a remote Kubernetes cluster on page 420](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane of the team space.

The **Agents and Intercepts** page is displayed.

You can view the agents, intercepts, Dockers, or clusters that are registered with the team space on Test Hub.

3. You can view the clusters that you own in any of the following ways:

- Search for the cluster by entering the name of the cluster in the **Search** field.



Note: You can enter either the full name or any text that is in the name. The search is enabled for case-sensitive text that you can enter.


- Sort the **Type** column to sort the items, and then identify the cluster by the name or owner.
- Sort the **Agents** column to sort the items, and then identify the cluster by the name or owner.

You can view the following details about the clusters that are in the team space:


- The projects in the team space to which the cluster is added, are displayed in the **Projects** column.
- The status of the cluster is displayed in the **Status** column.



Note: You can add clusters that are in the `Ready` state to your project, and then use them as locations to run tests.

4. Identify the cluster that you want to modify the configurations, and then click the **Edit** icon .



Note: The  icon is displayed in the **Actions** column in the row of the cluster only if you own the cluster.

The **Update cluster** dialog is displayed.

5. Perform the following steps to change the `kubeconfig` file:
 - a. Click **Browse**, and locate the `kubeconfig` file of the Kubernetes cluster that you want to use.
 - b. Select the `kubeconfig` file, and then click **Open**.

The name of the `kubeconfig` file that you selected is displayed along with the address of the Kubernetes cluster that is specified in the file.

- c. Click **Next**.
6. Perform the following steps to change the namespace or name of the cluster:
 - a. Edit or change the namespace of the cluster that you want to use, in the **Namespace** field.
 - b. Edit or change the name of the cluster in the **Name** field.
 - c. Click **Next**.

7. Perform the following steps to change the ingress details of the cluster:
 - a. Select the ingress type of the cluster from the **Ingress type** list, if you want to use a different cluster.
 - b. Enter or change the ingress host pattern of the cluster in the **Ingress host pattern** field, if you did not provide it when setting up the remote cluster or change it to match the cluster you want to use.
 - c. Enter the secret if you did not provide it when setting up the remote cluster.

The secret is used to authenticate the remote cluster to use the licenses from the Team space License Server so that the inbound traffic is allowed into the ingress domain.



Note: The secret to use must be available with your Test Hub server administrator.

- d. Click **Next**.
8. Optionally, perform the following steps to edit or change the code snippets for `nodeSelectors` and `tolerations`, if applicable for the remote cluster:

- a. Enter the code snippet for `nodeSelectors` in a `yaml` format in the **Node selectors** field.
 - b. Enter the code snippet for `tolerations` in a `yaml` format in the **Tolerations** field.
9. Click **Update**.

The remote cluster is displayed with the updated configurations.



Note: The projects that are associated with the cluster are displayed in the **Projects** column and you need not add the cluster to the same project again.

Results

You have updated an existing remote cluster with the changed configurations.

What to do next

You can select the updated cluster as an alternate location to run the test assets or resources that are in your project while you configure a run from the **Execution** page.

Related information

[Management of remote Kubernetes clusters on page 417](#)

Removing a remote Kubernetes cluster from a project

You can remove a remote Kubernetes cluster from a project in a team space on HCL DevOps Test Hub (Test Hub) when you no longer need the cluster to run tests.

Before you begin

You must have completed the following tasks:

1. Registered a remote Kubernetes cluster with your team space on Test Hub. See [Registering a remote Kubernetes cluster on page 420](#).
2. Added the Kubernetes cluster to a project in your team space. See [Adding a remote Kubernetes cluster to a project on page 425](#).



Note: You can remove remote Kubernetes clusters that only you added to the project.

3. Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
4. Been assigned a role as an *Owner* in the project.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project in your team space.

The **Overview** page is displayed.

3. Click **Infrastructure > Agents and Intercepts** in the navigation pane of your team space.

The **Agents and Intercepts** page is displayed.

You can view the agents, intercepts, Dockers, or clusters that are added to your project.


4. You can view the clusters that you own in any of the following ways:

- Search for the cluster by entering the name of the cluster in the **Search** field.




Note: You can enter either the full name or any text that is in the name. The search is enabled for case-sensitive text that you can enter.

- Sort the **Type** column to sort the items, and then identify the cluster by the name or owner.
- Sort the **Agents** column to sort the items, and then identify the cluster by the name or owner.

5. Identify the cluster that you want to remove from the project, and then click the **Remove cluster** icon .



Note: The  icon is displayed in the **Actions** column in the row of the cluster only if you own the remote cluster.

A message is displayed that the remote cluster is removed from the project successfully.

Results

You have removed the remote cluster from your project. The remote cluster is no longer available as a location when you configure a run of the test assets or resources that are in your project.

What to do next

You might have to add a remote cluster to your project to use the remote cluster as a location to run test assets or resources.

Related information

[Management of remote Kubernetes clusters on page 417](#)

[Unregistering a remote Kubernetes cluster from a team space on page 430](#)

[Editing configurations of a remote Kubernetes cluster on page 427](#)

Unregistering a remote Kubernetes cluster from a team space

You can unregister a remote Kubernetes cluster from a team space in HCL DevOps Test Hub (Test Hub) when you no longer require it.

Before you begin

You must have completed the following tasks:

1. Registered a remote Kubernetes cluster with your team space on Test Hub. See [Registering a remote Kubernetes cluster on page 420](#).
2. Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
3. Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane of the team space.

The **Agents and Intercepts** page is displayed.

You can view the agents, intercepts, Dockers, or clusters that are registered with the team space on Test Hub.

3. You can view the clusters that you own in any of the following ways:

- Search for the cluster by entering the name of the cluster in the **Search** field.





Note: You can enter either the full name or any text that is in the name. The search is enabled for case-sensitive text that you can enter.

- Sort the **Type** column to sort the items, and then identify the cluster by the name or owner.
- Sort the **Agents** column to sort the items, and then identify the cluster by the name or owner.

4. Identify the cluster that you want to remove.



Note: The  and  icons are displayed in the **Actions** column in the row of the cluster only if you own the remote cluster.

5. Unregister the cluster by completing the following steps:

- a. Click the **Unregister cluster** icon .
- b. Click **Unregister** in the **Unregister cluster** dialog box.

A message is displayed that the remote cluster is unregistered successfully.

Results

You have unregistered the remote cluster from a team space on Test Hub. You cannot run test assets or resources on this remote cluster if it is added to your project. You might have to add it again to your team space, if you want to use the remote cluster in your projects.

Related information

[Management of remote Kubernetes clusters on page 417](#)

[Editing configurations of a remote Kubernetes cluster on page 427](#)

[Adding a remote Kubernetes cluster to a project on page 425](#)

[Removing a remote Kubernetes cluster from a project on page 429](#)

Management of mobile device cloud credentials

You can find information about managing the credentials that are required to connect to mobile device clouds from HCL DevOps Test Hub (Test Hub).

Contents

- [Mobile device clouds on page 432](#)
- [Mobile device clouds and Test Hub on page 432](#)
- [Mobile device clouds that are supported on page 433](#)
- [Details required of mobile device clouds on page 433](#)
- [Prerequisites for using mobile devices on mobile device clouds on page 433](#)
- [Overview of tasks to manage mobile device clouds on page 434](#)

Mobile device clouds

Mobile device clouds are cloud computing services offered by cloud service providers who host real mobile devices in a cloud. You can purchase plans from the service providers to use the mobile devices in the cloud to test your mobile apps. The cloud service providers also offer different browsers in the cloud to test web applications.

To use the cloud services, you must subscribe to their services that suit your testing requirements.

Mobile device clouds and Test Hub

If you already use mobile device clouds to test your mobile apps, testers in your organization must be required to set up the connection to the mobile device clouds individually. They use the same credentials that you subscribed to as an organization. This can be a potential risk of exposure of your subscription details and without proper monitoring can lead to misuse of the credentials.

You can use Test Hub to connect to mobile device clouds. You must first configure a cloud credential by using the details of your cloud subscription account.

You can then add testers from your organization as authorized users who can use the credential to connect to mobile device clouds. Such members need not enter the subscription details to connect to the cloud. Only the authorized users of the credential can run their tests on the devices that are connected to the mobile device cloud. With this mechanism, you can prevent the exposure of the credentials of your subscription to testers and also prevent testers from creating separate connections to the same mobile device cloud.

Mobile device clouds that are supported

Test Hub supports connections to the following mobile device clouds:

- BitBar Cloud
- BrowserStack Cloud
- pCloudy Cloud
- Perfecto Mobile Cloud

Details required of mobile device clouds

You must ensure that you have the details and valid credentials of your subscription to the mobile device cloud that you want to connect and use. See the following table for the details required for the different mobile device clouds:

Attribute	Description	Mobile device cloud			
		BitBar Cloud	BrowserStack Cloud	pCloudy Cloud	Perfecto Mobile Cloud
Host	The hostname of the mobile device cloud instance.	Required	Required	Required	Required
API key	The API key of your cloud account to authenticate the connection.	Required	Required	Required	Not applicable
Security token	The token used to authenticate the connection.	Not applicable	Not applicable	Not applicable	Required
Project	The project name to use on the mobile device cloud.	Required	Required	Not applicable	Not applicable
Username	The username of your cloud account.	Not applicable	Required	Required	Not applicable

Prerequisites for using mobile devices on mobile device clouds

When you want to use the available mobile devices that are connected to a mobile device cloud, you must ensure that you have completed certain tasks. Refer to the following table for details of the tasks that must be completed before tests can be run on the mobile devices connected to mobile device clouds:

Device cloud	Testing on Android OS	Testing on iOS
BitBar Cloud	<ul style="list-style-type: none"> • Uploaded the <code>.apk</code> file of the Android application that you want to test. • Installed the Android application that you want to test on the Android device. 	<ul style="list-style-type: none"> • Installed the <code>.ipa</code> or <code>.app</code> file of the iOS application that you want to test. • Installed the iOS application that you want to test on the iOS device.
BrowserStack Cloud		
Perfecto Mobile Cloud		
pCloudy Cloud	<ul style="list-style-type: none"> • Uploaded the <code>.apk</code> file of the Android application that you want to test. • Installed the Android application that you want to test on the Android device. • Verified that the device you want to test is available and in the <i>Released</i> state on the pCloudy Cloud. 	<ul style="list-style-type: none"> • Uploaded the <code>.ipa</code> file of the iOS application that you want to test. • Installed the iOS application that you want to test on the iOS device. • Verified that the device you want to test is available and in the <i>Released</i> state on the pCloudy Cloud.

Overview of tasks to manage mobile device clouds

You as a *Team Space Owner* can perform the following tasks:

- Configure the connection to mobile device clouds. You must create a *cloud credential* in the team space that represents a connection to a mobile device cloud. You must use the details of your mobile device cloud subscription account to configure the cloud credential to the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).
- Add your team space members to access the mobile device cloud. The added members become the authorized users of the cloud credential, and then can connect to the mobile device cloud from Test Hub. See [Adding authorized users to a cloud credential on page 438](#).
- View the details of the cloud credential that you added to the team space. See [Viewing details of cloud credentials on page 439](#).
- Remove the authorized users of the cloud credential, when you want to reduce the number of users accessing the mobile device cloud. See [Removing authorized users from a cloud credential on page 444](#).
- Edit an existing cloud credential that you configured to modify the name of the credential, project, or username of your cloud subscription account. See [Editing details of cloud credentials on page 442](#).
- Delete an existing cloud credential that you configured, from the team space. You might want to delete the credential when your cloud subscription to the mobile device cloud has expired and you no longer need the connection. See [Deleting an existing cloud credential on page 445](#).
- Export details of all existing cloud credentials that are configured in your team space, when you might want to use the details elsewhere. You can export details such as the name of the credentials, the credential type, and the email ID of authorized users of the credentials. See [Viewing details of cloud credentials on page 439](#).

You as a *Project Owner* or *Tester* can perform the following tasks on cloud credentials or the mobile device clouds:

- Ensure that you are added as an authorized user of a cloud credential in the team space. See [Viewing details of cloud credentials on page 439](#).
- Configure the following mobile tests to run on the mobile devices that are connected to mobile device clouds:
 - [Configuring an AFT Suite to run on a mobile device by using a cloud credential on page 478](#).
 - [Configuring a run of a Compound Test that contains mobile tests to run by using a cloud credential on page 523](#).

Adding cloud credentials of mobile device clouds

You as a *Team Space Owner* can add a cloud credential in the team space by using the details of your mobile device cloud subscription. This cloud credential represents a connection between HCL DevOps Test Hub (Test Hub) and a mobile device cloud.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you have the details of your subscription and valid credentials of the mobile device cloud to which you want to connect. See [Details required of mobile device clouds on page 433](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Cloud Credentials** on the navigation pane.

The **Cloud Credentials** page is displayed.

3. Click **Add cloud credential**.

The **Details** tab of the **Add cloud credential** dialog is displayed.

4. Perform the following steps in the **Details** tab:

- a. Enter a name in the **Name** field so that you can identify the mobile device cloud, which is associated with a specific subscription account.
- b. Select the mobile device cloud from the **Cloud Authentication** list.
- c. Click **Next**.

The **Authentication** tab of the **Add cloud credential** dialog is displayed.

5. Enter the relevant details of your subscription account that you have in the mobile device cloud, in the **Authentication** tab.




You can find information about the details that are required for each of the supported mobile device clouds and the action you need to perform for the required fields in the following table:













Field	Mobile device cloud				Action
	BitBar Cloud	BrowserStack Cloud	pCloudy Cloud	Perfecto Mobile Cloud	
Host	Required	Required	Required	Required	Enter the hostname of the mobile device cloud instance.
API key	Required	Required	Required	Not applicable	Enter the API key of your cloud subscription account to authenticate the connection.
Security token	Not applicable	Not applicable	Not applicable	Required	Enter the token used to authenticate the connection.
Project	Required	Required	Not applicable	Not applicable	Enter or select the project to use that is in the mobile device cloud.
Username	Not applicable	Required	Required	Not applicable	The username of your cloud subscription account.

6. Click **Add**.

The cloud credential for your specific subscription account with a mobile device cloud is displayed in a table on the **Cloud Credentials** page.

You can find the following details about the cloud credential on the **Cloud Credentials** page.

Column header	Details displayed
Provider	<p>The name of the mobile device cloud service provider.</p> <p>You can view the Toggle icon  associated with the service providers when you add a cloud credential.</p> <p> Note: If you are the owner of the cloud credential, you can switch to an On/Off position. However, when you are not the owner and you try to use the Toggle icon  an error message is displayed.</p>
Name	The name you entered for the cloud credential when you created the connection in the team space.

Column header	Details displayed															
Credential Type	The type of cloud that is connected to Test Hub. Displays <i>Device Cloud</i> when you create a credential to connect to a mobile device cloud.															
Members	The abbreviated names of the team space members who are authorized users of the cloud credential. The abbreviated name is the combination of the first letters of the first name and last name.															
Actions	<p>The icons for the different actions that you can perform on the credential. Refer to the following table for the icons and the actions:</p> <table border="1"> <thead> <tr> <th>Icon</th> <th>Action</th> <th>Available for</th> </tr> </thead> <tbody> <tr> <td></td> <td>Click to view details of the credential of the cloud subscription account.</td> <td>Team space owner and authorized user</td> </tr> <tr> <td></td> <td>Click to edit and update the credential for any of your cloud subscription details.</td> <td>Team space owner</td> </tr> <tr> <td></td> <td>Click to remove the configured connection to the mobile device cloud account from the server.</td> <td>Team space owner</td> </tr> <tr> <td></td> <td>Click to add team space members as authorized users of the credential.</td> <td>Team space owner</td> </tr> </tbody> </table>	Icon	Action	Available for		Click to view details of the credential of the cloud subscription account.	Team space owner and authorized user		Click to edit and update the credential for any of your cloud subscription details.	Team space owner		Click to remove the configured connection to the mobile device cloud account from the server.	Team space owner		Click to add team space members as authorized users of the credential.	Team space owner
Icon	Action	Available for														
	Click to view details of the credential of the cloud subscription account.	Team space owner and authorized user														
	Click to edit and update the credential for any of your cloud subscription details.	Team space owner														
	Click to remove the configured connection to the mobile device cloud account from the server.	Team space owner														
	Click to add team space members as authorized users of the credential.	Team space owner														

Results

You have added credentials of mobile device cloud accounts so that Test Hub can connect to the mobile device clouds.

What to do next

You can add team space members as authorized users of the credentials. See [Adding authorized users to a cloud credential on page 438](#).

Related information

[Management of mobile device cloud credentials on page 432](#)

Adding authorized users to a cloud credential

After you add the mobile device cloud account details to a cloud credential in your team space in HCL DevOps Test Hub (Test Hub), you can add other team space members as authorized users of this cloud credential. The authorized users can then run mobile tests on the devices that are connected to the mobile device cloud by using this cloud credential. The authorized users need not configure a connection to the mobile device cloud.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Added the cloud credentials of your mobile device cloud account to configure the connection between Test Hub and the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).

About this task

You as a *Team Space Owner* can add the cloud credentials of your mobile device cloud account to your team space. You can then add team space members to this cloud credential as authorized users. The authorized users can use the cloud credential to connect to the mobile device cloud from Test Hub. They can configure their mobile tests in any project in the team space to run on the mobile devices that are connected to the mobile device cloud.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Cloud Credentials** on the navigation pane.

The **Cloud Credentials** page is displayed with the cloud credentials that are added to the team space.

3. Identify the cloud credential by performing any of the following actions:

- Scroll through the table of cloud credentials to identify the cloud credential with the name you provided.
- Search for the cloud credential by entering the name of the credential in the **Search** field.
- Sort the **Provider** column so that the names of the device cloud providers are listed alphabetically in either increasing or decreasing order. Identify your device cloud provider and look for the name of your credential.

You identified the cloud credential that you want to use.

4. Perform the following steps to add team space members as authorized users:

- a. Click the **Add Members** icon  in the row of the credential.

The **Add New Member** dialog is displayed.

- b. Enter any text that is contained in the name or email ID of the team space member that you want to add in the field.

The team space users are listed whose names or email ID matches the search text.

- c. Select the team space member.

The selected member is added to the list of authorized users and is displayed under **Members with access**.

- d. Click **Close** to close the dialog.

The abbreviated name of the team space member is displayed under the **Members** column in the row of the cloud credential.



Note: The abbreviated name is the combination of the first letters of the first name and last name.

You can view the first name, last name, username, and the email ID of the added member when you hover over the abbreviated name.

Results

You as a *Team Space Owner* added team space members as authorized users of the cloud credential.

What to do next

You can inform the team space members that they are added as authorized users of the cloud credential. The authorized users can use the credential to connect and run tests on mobile device clouds. The authorized users of the credential can configure the following test runs:

- [Configuring an AFT Suite to run on a mobile device by using a cloud credential on page 478.](#)
- [Configuring a run of a Compound Test that contains mobile tests to run by using a cloud credential on page 523.](#)

Related information

[Management of mobile device cloud credentials on page 432](#)

Viewing details of cloud credentials

If you are a *Team Space Owner* in HCL DevOps Test Hub (Test Hub), you can view the details of the cloud credential when you want to decide your action for adding or removing team space members, changing your cloud subscription details, or deleting a credential. If you are a *Project Owner* or *Tester*, you can verify whether you are added as an authorized user to use the cloud credential.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* in the team space, a *Project Owner*, or a *Tester* in the project. See [Managing members and their roles in a team space on page 1006](#).
- Added the cloud credentials of your mobile device cloud account to configure the connection between Test Hub and the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).
- Added team space members as authorized users of the cloud credential. See [Adding authorized users to a cloud credential on page 438](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Cloud Credentials** on the navigation pane.

The **Cloud Credentials** page is displayed with the cloud credentials that are added to the team space.

3. Identify the cloud credential by performing any of the following actions:

- Scroll through the table of cloud credentials to identify the cloud credential with the name you provided.
- Search for the cloud credential by entering the name of the credential in the **Search** field.
- Sort the **Provider** column so that the names of the device cloud providers are listed alphabetically in either increasing or decreasing order. Identify your device cloud provider and look for the name of your credential.

You identified the cloud credential that you want to use.

4. Select the action that you want to perform:

- Verify whether you are added as an authorized user of the cloud credential by the *Team Space Owner*, if you are a *Project Owner* or *Tester*, go to [Step 5 on page 440](#).
- View details of the cloud credential, if you are a *Team Space Owner*, *Project Owner*, or *Tester*, go to [Step 6 on page 441](#).
- Export the details of all the cloud credentials in your team space, if you are a *Team Space Owner*, go to [Step 7 on page 442](#).

5. Perform the following steps to verify whether you are added as an authorized user of the cloud credential:

- a. Look for your abbreviated name under the **Members** column in the row of the identified cloud credential.



Note: The abbreviated name is the combination of the first letters of the first name and last name.


- b. Hover over the abbreviated name to view the first name, last name, username, and the email ID to verify that they are your details.

You have verified whether you are an authorized user of the cloud credential.



Note: If you do not find your abbreviated name or details, you need to contact your team space owner to add you as an authorized user.

c. Go to [Results on page 442](#).


6. Click the **View** icon  in the row of the credential that you want to view.

The details of the cloud credential are displayed in the **View cloud credential** dialog.

You can view the following details about the credential for the mobile device cloud that was configured in the cloud credential:

Attribute	Displays	Mobile device cloud details displayed			
		BitBar Cloud	Browser-Stack Cloud	pCloudy Cloud	Perfec-to Mobile Cloud
Owner	The username and email ID of the team space owner who configured the cloud credential.	✓	✓	✓	✓
Credential name	The name provided to the cloud credential by the team space owner while the connection with the mobile device cloud was configured.	✓	✓	✓	✓
Type	The name of the mobile device cloud to which the connection was configured in the cloud credential.	✓	✓	✓	✓
Host	The hostname or URL of the mobile device cloud instance.	✓	✓	✓	✓
Project	The name of the project on the mobile device cloud.	✓	✓	✗	✗
Username	The username of your cloud subscription account.	✗	✓	✓	✗
API key	The API key of your cloud subscription account to authenticate the connection.	✓	✓	✓	✗
Security token	The token used to authenticate the connection.	✗	✗	✗	✓
Access to	The team space members that the team space owner added as authorized users of the cloud credential.	✓	✓	✓	✓

You can close the dialog by clicking **Close**.

7. Perform the following steps to export details of all existing cloud credentials that are configured in your team space when you want to use the details elsewhere, such as the name of the credentials, the credential type, and email ID of authorized users of the credentials members:
 - a. Click the **Export** icon .
 - b. Click the **CSV** option.
 - c. Enter a name for the file.
 - d. Select a location on your computer to save the file.
 - e. Click **Save**.

You saved the details of all the credentials in your team space as a CSV file to your computer.

Results

You have completed the following tasks:

- If you are a *Project Owner* or *Tester*, you verified whether you are an authorized user of the cloud credential.
- If you are a *Team Space Owner*, *Project Owner*, or *Tester*, you have viewed the details of the cloud credential that you added.
- If you are a *Team Space Owner*, you saved the details of all cloud credentials that are in your team space to a file on your computer.

Related information

[Management of mobile device cloud credentials on page 432](#)

Editing details of cloud credentials

If any of the details of your cloud subscription account change, as a *Team Space Owner* in HCL DevOps Test Hub (Test Hub) you can view the details of the existing cloud credential and edit it to update the credential with the latest details.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Added the cloud credentials of your mobile device cloud account to configure the connection between Test Hub and the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).

1. Log in to Test Hub.


The team space that contains your project is displayed.

- Click **Infrastructure > Cloud Credentials** on the navigation pane.

The **Cloud Credentials** page is displayed with the cloud credentials that are added to the team space.

- Identify the cloud credential by performing any of the following actions:
 - Scroll through the table of cloud credentials to identify the cloud credential with the name you provided.
 - Search for the cloud credential by entering the name of the credential in the **Search** field.
 - Sort the **Provider** column so that the names of the device cloud providers are listed alphabetically in either increasing or decreasing order. Identify your device cloud provider and look for the name of your credential.

You identified the cloud credential that you want to use.

- Click the **Edit cloud credential** icon  in the row of the credential that you want to edit.

The **Update cloud credential** dialog is displayed.

- Modify the name of the cloud credential in the **Details** tab, and then click **Next**.

The **Authentication** tab of the **Update cloud credential** dialog is displayed.

- Modify relevant fields of your subscription account if they have changed, in the **Authentication** tab.

You can find information about the details that are required for each of the supported mobile device clouds and the action you need to perform for the relevant and required fields in the following table:

Field	Mobile device cloud				Action
	BitBar Cloud	BrowserStack Cloud	pCloudy Cloud	Perfecto Mobile Cloud	
Host	Required	Required	Required	Required	Enter the hostname of the mobile device cloud instance.
API key	Required	Required	Required	Not applicable	Enter the API key of your cloud subscription account to authenticate the connection.
Security token	Not applicable	Not applicable	Not applicable	Required	Enter the token used to authenticate the connection.
Project	Required	Required	Not applicable	Not applicable	Enter or select the project to use that is in the mobile device cloud.
Username	Not applicable	Required	Required	Not applicable	The username of your cloud subscription account.

7. Click **Update**.

The existing cloud credential is updated with the changed details of your subscription account with a mobile device cloud.

Results

You edited an existing cloud credential in your team space and updated it with the changed details of your cloud subscription account.

Related information

[Management of mobile device cloud credentials on page 432](#)

Removing authorized users from a cloud credential

When you want to restrict the number of users accessing the mobile device clouds by using your subscription account due to any change in your subscription fee or for other valid reasons, you might want to remove some of the authorized users of the cloud credential in HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Added the cloud credentials of your mobile device cloud account to configure the connection between Test Hub and the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).
- Added team space members as authorized users of the cloud credential. See [Adding authorized users to a cloud credential on page 438](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Cloud Credentials** on the navigation pane.

The **Cloud Credentials** page is displayed with the cloud credentials that are added to the team space.

3. Identify the cloud credential by performing any of the following actions:

- Scroll through the table of cloud credentials to identify the cloud credential with the name you provided.
- Search for the cloud credential by entering the name of the credential in the **Search** field.
- Sort the **Provider** column so that the names of the device cloud providers are listed alphabetically in either increasing or decreasing order. Identify your device cloud provider and look for the name of your credential.


You identified the cloud credential that you want to use.

4. Perform the following steps to remove authorized users of the cloud credential:

- a. Click the **Add Members** icon  in the row of the credential.

The **Add New Member** dialog is displayed.

- b. Identify the member from the listed members displayed under **Members with access**.

- c. Click  in the row of the identified member.

The selected member is removed from the list displayed under **Members with access**.

- d. Click **Close** to close the dialog.

The abbreviated name of the authorized user is removed from the display under the **Members** column in the row of the cloud credential.

Results

You as a *Team Space Owner* removed authorized users from the cloud credential.

What to do next

You can inform the team space members that they are removed as authorized users of the cloud credential.

Related information

[Management of mobile device cloud credentials on page 432](#)

Deleting an existing cloud credential

When your cloud subscription to the mobile device cloud expires and you no longer need the connection from HCL DevOps Test Hub (Test Hub), you can delete an existing cloud credential that you configured in your team space.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Added the cloud credentials of your mobile device cloud account to configure the connection between Test Hub and the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).

1. Log in to Test Hub.


The team space that contains your project is displayed.

2. Click **Infrastructure > Cloud Credentials** on the navigation pane.

The **Cloud Credentials** page is displayed with the cloud credentials that are added to the team space.

3. Identify the cloud credential by performing any of the following actions:
 - Scroll through the table of cloud credentials to identify the cloud credential with the name you provided.
 - Search for the cloud credential by entering the name of the credential in the **Search** field.
 - Sort the **Provider** column so that the names of the device cloud providers are listed alphabetically in either increasing or decreasing order. Identify your device cloud provider and look for the name of your credential.

You identified the cloud credential that you want to use.

4. Click the **Delete cloud credential** icon  in the row of the credential that you want to delete.

The **Delete cloud credential** dialog is displayed.

5. Click **Delete**.

The existing cloud credential is deleted and removed from the display on the **Cloud Credentials** page.

Results

You deleted an existing cloud credential that you previously configured, from your team space.

Related information

[Management of mobile device cloud credentials on page 432](#)

Test run configurations

You can configure and run tests in HCL DevOps Test Hub (Test Hub) after you add the test resources to your project in your team space.


Before you configure a test run, you must have completed the following tasks:


- Created a project on Test Hub or you must have been granted access to a project with the *Tester* role assigned. See [Managing DevOps Test Hub projects on page 1010](#).
- Created tests in the desktop clients and committed the test assets and test resources to a remote repository. You must have added the remote repository to the project.
- Read the considerations that you must take into account for certain types of tests. See [Prerequisites to running tests on page 371](#).

You must be a project *Owner* or a member with the *Tester* role assigned to configure and run a test.

You can find information about how to configure and run the following types of test assets:

Type of Suite or test	Suite or test	More information
AFT Suite	Suite	<ul style="list-style-type: none"> • Configuring an AFT Suite run on page 453 • Configuring an AFT Suite to run on a mobile device by using a cloud credential on page 478
API Suite	Suite	<ul style="list-style-type: none"> • Configuring an API Suite or an API test run on page 486
Compound Test containing HTML tests	Suite	<ul style="list-style-type: none"> • Configuring a run of a Compound Test that contains HTML tests on page 498
Compound Test containing Web UI tests	Suite	<ul style="list-style-type: none"> • Configuring a run of a Compound Test that contains Web UI tests on page 533 • Configuring a run of a Compound Test that contains Web UI tests to run by using variables on page 545 • Configuring a run of a Compound Test that contains Web UI tests to run by using a cloud credential on page 559
Compound Test containing mobile tests	Suite	<ul style="list-style-type: none"> • Configuring a run of a Compound Test that contains mobile tests to run by using variables on page 509 • Configuring a run of a Compound Test that contains mobile tests to run by using a cloud credential on page 523
Compound Test containing performance tests	Suite	<ul style="list-style-type: none"> • Configuring a run of a Compound Test that contains performance tests on page 569

Type of Suite or test	Suite or test	More information
JMeter test	Suite	<ul style="list-style-type: none"> • Configuring a run of a JMeter test on page 665
JUnit test	Suite	<ul style="list-style-type: none"> • Configuring a run of a JUnit test on page 674
Postman test	Suite	<ul style="list-style-type: none"> • Configuring a run of a Postman test on page 683
Rate Schedule	Suite	<ul style="list-style-type: none"> • Configuring a run of a Rate Schedule or VU Schedule on page 692
Selenium test	Suite	<ul style="list-style-type: none"> • Configuring a run of a Selenium test on page 704
VU Schedule	Suite	<ul style="list-style-type: none"> • Configuring a run of a Rate Schedule or VU Schedule on page 692
API test	Test	<ul style="list-style-type: none"> • Configuring an API Suite or an API test run on page 486
Functional test  Note: The Functional test can contain any of the following individual tests:	Test	<ul style="list-style-type: none"> • Configuring a run of a mobile test to run by using variables on page 581 • Configuring a run of a mobile test to run on a mobile device by using a cloud credential on page 595

Type of Suite or test	Suite or test	More information
 <ul style="list-style-type: none"> • Mobile test • Web UI test • Windows test 		<ul style="list-style-type: none"> • Configuring a run of a Web UI test on page 617 • Configuring a run of a Web UI test to run by using variables on page 629 • Configuring a run of a Web UI test to run by using a cloud credential on page 643 • Configuring a run of a Windows test on page 653
Performance test	Test	<ul style="list-style-type: none"> • Configuring a run of a Performance test on page 605

Configuring a Suite or test for a quick run

You can configure any type of Suite or test to be run on HCL DevOps Test Hub (Test Hub) when you want to quickly ensure that the test runs correctly. You might not want to set the different options for the test nor want to schedule the run.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Prerequisites to running tests on page 371](#) if they apply to the test you want to configure for a run.
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.

The test assets in the selected tab are displayed.


5. Identify the test asset that you want to run by performing any of the following actions:





































- Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.

- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path
- Last Result
- Next Run
- Components

- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run, if you want to run a different version other than the latest version.
8. No action is required if you want to initiate the test run immediately after you click **Execute**.
9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page.


10. Click **Execute**.
The test run is initiated.

Results

You have started a test run.

What to do next

You can perform any of the following tasks::

- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Viewing the progress of running test assets on page 731](#)

[Stopping test runs on page 736](#)

[Canceling scheduled test runs on page 739](#)

Configuring an AFT Suite run

After you added the test resources that you created in the desktop client to the project, you can configure an AFT Suite to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Read [Test run considerations for AFT Suites or Functional tests on page 371](#), if you want to configure a run for an AFT Suite that has the agent location configured in the `AFT.XML` file.
- Read and completed the tasks mentioned in [Test run considerations for browser-based Web UI tests on page 376](#), if you want to run tests that use Chrome or Edge browsers.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.






















The test assets in the selected tab are displayed.


5. Identify the test asset that you want to run by performing any of the following actions:



















- Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.





Important: Click **Execute** only after you have configured the other settings in this dialog.




Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:

Schedule	Actions
A single test run	Perform the following steps: <ol style="list-style-type: none"> a. Click the Calendar icon  in the row of the Start field. b. Select the date. c. Select the time at which the test run must start.


Schedule	Actions
	<p>d. Select the Never option in the Repeat every field.</p> <p>e. Click Save.</p> <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p> <div data-bbox="708 1283 1421 1444" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Repeat every <input style="width: 80px;" type="text" value="1"/> <input style="width: 150px;" type="text" value="Week(s)"/> ▾</p> <p style="text-align: center;"> <input type="radio"/> S <input checked="" type="radio"/> M <input type="radio"/> T <input checked="" type="radio"/> W <input type="radio"/> T <input checked="" type="radio"/> F <input type="radio"/> S </p> </div> <ol style="list-style-type: none"> Set the option to end the recurring test runs from the following options:

Schedule	Actions
	<ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

If you are running an AFT Suite and the following conditions are true, then you must perform the next step:

- You do not want to run the test on the agent configured in the AFT XML file.
- You have not selected the agent from the **Override** column in the **Location** tab.

10. Click **Advanced** to make the following advanced configurations:

- a. Add the following setting in the **Program Arguments** field:

```
-swaplocation <configured_agent_location>:<overriding_agent_location>
```

For example, if the configured agent location is *91.2.352.24* and the remote agent where you want to run the test is *100.35.117.164*, then the entry in the **Program Arguments** field is as follows:

```
-swaplocation 91.2.352.24:100.35.117.164
```

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

a. Click the **DATA SOURCES** tab, if it is not already open.

b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.




Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.



 **Important:** If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.


If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

a. Click the **VARIABLES** tab, if it is not already open.

b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.

 **Note:** You must have created a file with the variables before you can select the file.

The default value for the variables is null or an empty field.


If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.


b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. The Select issue field is displayed. <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.

Export options	Results in...	Actions
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Select any of the following steps that depends on the location to run the test:
 - If you want to run the tests on the remote agents, go to [Step 15 on page 463](#).
 - If you want to run the tests on a Kubernetes cluster, go to [Step 16 on page 464](#).
15. Follow the instructions if you want to run a test on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:
 - a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the performance agent is displayed.

 **Note:** You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

- b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other perfor-	The performance agent that is specified in the test assets is	Select the performance agent in the Host that is also displayed

If...	Then...	Action
mance agents are added to the project.	displayed in the Override column.	in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the agent.
The agent that is specified in the test assets is available and other agents added to the project.	Performance agents that have capabilities that match with the agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. You can find details of both performance agents for the matching capabilities when you hover over the performance agent in the Override column.	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.


c. Click **Execute**.

The test run is initiated.

16. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

 **Important:** You must have added the following remote hosts to your project that are then displayed under the **Override** column:

- Docker hosts that are registered with Test Hub.
- Kubernetes clusters that are registered with a team space.

 **Notes:**

- If the remote Docker hosts or clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote Docker hosts or clusters are added to your project, the added Docker hosts or clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts or clusters are available in your project.
- Select the remote Docker host or cluster from the list when a remote Docker host or cluster is available in your project.
- Select **No override options**, if you selected any remote Docker host or cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of an AFT Suite.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring an AFT Suite to run on a mobile device by using variables

After you added the test resources that you created in the desktop client to the project, you can configure an AFT Suite to run on a mobile device from HCL DevOps Test Hub (Test Hub). You must specify the variables for the mobile tests to run on mobile devices that are connected to an agent or a device cloud.

Before you begin

You must have completed the following tasks:

- Read [Test run considerations for AFT Suites or Functional tests on page 371](#), if you want to configure a run for an AFT Suite that has the agent location configured in the `AFT XML` file.
- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Created a variable file if you want to import the variables file. The file must contain the details of the UI Test agent or the device cloud to which the mobile device is connected.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Ensured that you are a member of the cloud credential that is configured to connect to the mobile device cloud. See [Viewing details of cloud credentials on page 439](#).

About this task

You can run an AFT Suite, which contains Web UI or mobile tests to run on a mobile device, emulator, or simulator that is connected to an agent or a device cloud. You must create the tests in Test UI, and then add the test asset to your project on Test UI. You can then run the tests on the mobile device, emulator, or simulator that is connected to any of the following agents or clouds:

- Remote UI Test agents. Requires the variables at test run time as specified in [Variables for the UI Test agent on page 467](#).
- The BitBar Cloud. Requires the variables at test run time as specified in [Table 23: Variables for the BitBar Cloud on page 467](#).
- The BrowserStack Cloud. Requires the variables at test run time as specified in [Table 24: Variables for the BrowserStack Cloud on page 468](#).
- The pCloudy Cloud. Requires the variables at test run time as specified in [Table 25: Variables for the pCloudy Cloud on page 468](#).
- The Perfecto Mobile Cloud. Requires the variables at test run time as specified in [Table 26: Variables for the Perfecto Mobile Cloud on page 469](#).

You must provide the details of the UI Test agent or the mobile cloud to which the mobile devices are connected as variables in [Step 10 on page 475](#). You can either enter the variables or use the file in which you entered the variables.

You must refer to the following tables for the variables that are required for a successful run:

Table 22. Variables for the UI Test agent

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the UI Test agent. For example, <i>Emulator:Pixel_3a-API_33</i> .
appium.server.host	Specify the hostname or IP address of the UI Test agent.
appium.server.port	Specify the port on the UI Test agent that is configured to communicate with Test Hub.

Table 23. Variables for the BitBar Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the BitBar Cloud. For example, <i>BitBar Cloud:Google Pixel 7 Pro</i> .
bitbar.apikey	Specify the user token generated for your BitBar Cloud account to authenticate your connection with the BitBar Cloud.

Table 23. Variables for the BitBar Cloud (continued)

Name of the Variable	Action for the Value field
bitbar.host	Specify the hostname of the BitBar Cloud instance.
bitbar.project	Specify the name of the project that contains the recorded test.
bitbar.testrun	Specify a name for the test run that must be displayed in the BitBar Cloud dashboard for the test run.

Table 24. Variables for the BrowserStack Cloud


Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the BrowserStack Cloud. For example, <i>BrowserStack Cloud:Google Pixel 7 Pro</i> .
browserstack.host	Specify the URL of the BrowserStack Cloud instance.
browserstack.username	Specify the user name of your BrowserStack Cloud account.
browserstack.apikey	Specify the API key of your BrowserStack Cloud account to authenticate your connection with the BrowserStack Cloud.
browserstack.project	Specify the name of the project that contains the recorded test.  Note: This variable is optional.

Table 25. Variables for the pCloudy Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the pCloudy Cloud. For example, <i>pCloudy Cloud:Google Pixel 7 Pro</i> .

Table 25. Variables for the pCloudy Cloud (continued)

Name of the Variable	Action for the Value field
pcloudy.apikey	Specify the API key that is generated for your pCloudy Cloud account to authenticate your connection with the pCloudy Cloud.
pcloudy.username	Specify the username with which you can access the pCloudy Cloud instance.
pcloudy.host	Specify the hostname of the pCloudy Cloud instance.

Table 26. Variables for the Perfecto Mobile Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the Perfecto Mobile Cloud. For example, <i>Perfecto Mobile Cloud:Google Pixel 7 Pro</i> .
perfecto.securitytoken	Specify the user token generated for your Perfecto Mobile Cloud account to authenticate your connection with the Perfecto Mobile Cloud.
perfecto.host	Specify the hostname of the Perfecto Mobile Cloud instance.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.


Tab	Description
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.










4. Click the **SUITES** or the **ADVANCED** tab.


5. Identify the test asset that you want to run by performing any of the following actions:










- Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test		✓	✓
	Functional test		✓	✓
	Performance test		✓	✓
	AFT Suite	✓		✓
	API Suite	✓		✓
	Compound Test	✓		✓
	HCL® AppScan CodeSweep	✓		✓
	JMeter Test	✓		✓
	JUnit Test	✓		✓
	Postman test	✓		✓



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.

- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the **✕** icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the **✕** icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	Perform the following steps: <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save.

Schedule	Actions
	<p>The test run is scheduled to run at the selected time on the scheduled date.</p>
<p>Recurring test runs</p>	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> Minute Hour Day Week Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p> <div data-bbox="708 1178 1421 1339" data-label="Image"> </div> Set the option to end the recurring test runs from the following options: <ul style="list-style-type: none"> Select the Never option, if you do not want the recurring test runs to end. Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. Click Save. <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>



9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.


10. Perform the following steps to provide the variables that specify the device cloud to which the mobile device is attached. You can either enter the variables that must be passed to the test at the test run time or import the file that contains the variables.
- a. Click the **VARIABLES** tab, if it is not already open.
 - b. Choose one of the following methods to add the variables:
 - To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
 - To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.



Note: You must have created a file with the variables before you can select the file.

11. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:
- a. Click the **RESULTS** tab.
 - b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you

Export options	Results in...	Actions
	selected issue that exists in your Xray project in Jira.	<p>can enter the issue key in the Jira Xray project.</p> <p>iii. Select the issue to which you want to export the test results.</p> <p>iv. Continue with the next step.</p>
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	<p>Perform the following steps:</p> <p>i. Select the New Jira item option.</p> <p>ii. Continue with the next step.</p>
None	<p>The test results are not exported.</p> <p> Note: This is the default option.</p>	<p>Perform the following steps when:</p> <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <p>i. Select the None option.</p> <p>ii. Continue with the next step.</p>

12. Perform any of the following actions that depend on whether you are using a remote performance agent or a mobile device cloud:

- Go to [Step 13 on page 476](#), if you are using an agent.
- Go to [Step 14 on page 478](#), if you are using a mobile device cloud.

13. Follow the instructions if you want to run the tests on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:

- a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability and capabilities of the agent are displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the performance agent.
The performance agent that is specified in the test assets is available and there are other performance agents added to the project.	Performance agents that have capabilities that match with the performance agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. You can find details of both performance agents for the matching capabilities when you hover over the agent in the Override column.	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

c. Click **Execute**.

14. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of an AFT Suite to run on a device, emulator, or simulator that is connected to an agent or mobile device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Test run configurations on page 446](#)

[Management of mobile device cloud credentials on page 432](#)

[Resetting the configuration settings for a test run on page 735](#)

Configuring an AFT Suite to run on a mobile device by using a cloud credential

After you added the test resources that you created in the desktop client to the project, you can configure an AFT Suite to be run on a mobile device by using a cloud credential from HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).

- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created a cloud credential in your team space that is configured to connect to the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).
- Ensured that you are a member of the cloud credential that is configured to connect to the mobile device cloud. See [Viewing details of cloud credentials on page 439](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.


4. Click the **SUITES** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
- Scroll through the list.











































Tip: You can hover over the icon in the **Type** column to know the type of the test asset.




Note: You can also identify the type of the asset from the icon that represents the test type as shown in the following table:



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:

- a. Click **New filter**.
- b. Enter a name for the filter.
- c. Select an operator, and then add a rule or a group of rules.
- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="708 264 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input type="text" value="1"/> <input type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Perform the following steps in the **ENVIRONMENT** tab:

a. Select the cloud credential of the mobile device cloud from the **Device cloud credentials** list.

The list displays the cloud credentials that are configured in the team space. The name of the cloud credential is appended to the name of the mobile device cloud in the list. For example, if the mobile device cloud provider is *BitBar* and the name of the credential is *BB_Account1*, then the name displayed in the list is *Bit Bar - BB_Account1*.



Note: Access to credentials for mobile device cloud services depends on the status of the provider. If you disable a provider on the **Cloud Credentials** page, you cannot access that device cloud provider from the **Device cloud credentials** list.

The devices that are available for testing on the selected mobile device cloud are not displayed because the devices are already specified in the AFT Suite.



Note: If you are connected to either the BitBar Cloud or BrowserStack Cloud, the **Projects** field is displayed.

- b. Select the project on the mobile device cloud that you want to use or create a project in the **Projects** field. The project overrides the project that is configured in the AFT Suite.

11. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of an AFT Suite to run on a mobile device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Test run configurations on page 446](#)

[Management of mobile device cloud credentials on page 432](#)

[Resetting the configuration settings for a test run on page 735](#)

Configuring an API Suite or an API test run

After you added the test resources that you created in the desktop client to the project, you can configure an API Suite or an API test to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Prerequisites to running tests on page 371](#), if they apply to the test that you want to configure for a run.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the following tasks if you are running API Suites or tests that use a transport and the transport requires third-party application `JAR` files for a successful run:
 - Read [Test run considerations for API Suites or API tests on page 373](#).
 - Copied the third-party application `JAR` files to the third-party application folder on the computer where Test Hub is installed, if you are running the API Suite in Kubernetes. See [Copying third-party application Jars to Kubernetes on page 160](#).
 - Copied the third-party application `JAR` files to the third-party application folder on the computer where the remote Docker host is installed, if you are running the API Suite on a remote Docker host. See [Copying third-party application Jars to a remote Docker host on page 406](#).
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.




The test assets in the selected tab are displayed.


5. Identify the test asset that you want to run by performing any of the following actions:





































- Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.

- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path
- Last Result
- Next Run
- Components

- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday</i>, <i>Wednesday</i>, and <i>Friday</i>:</p>

Schedule	Actions
	<div data-bbox="711 262 1421 420" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

Perform the following actions, if applicable for your test:

- Configuring test log reporting.


If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- Overriding results database that is configured in the API Suite project.



Note: You can use any of the databases that are specified in the system requirements as supported databases in Test Integrations and APIs.

Enter the following arguments in the **Program Arguments** field, if you want to override the **Results Database**:

Program Arguments	Description
<code>-resultDatabaseURL <database_URL></code>	Specifies the URL of the results database that overrides the results database that is configured for the project in Test Integrations and APIs.
<code>-resultDatabaseDriver <driver_name></code>	Specifies the driver of the results database that overrides the results database that is configured for the project in Test Integrations and APIs. For example, the driver for a MySQL database is <code>com.mysql.cj.jdbc.Driver</code> .
<code>-resultDatabaseUser- name <user_name></code>	Specifies the username that is authorized to access the results database that overrides the results database that is configured for the project in Test Integrations and APIs.
<code>-resultDatabasePa- ssword <encrypted_ password></code>	Specifies the password that is to be used to authenticate the user to access the results database that overrides the results database that is configured for the project in Test Integrations and APIs.  Note: You can encrypt the password by using the Encrypt-Password application that is available in the installation directory of Test Integrations and APIs.

For example, if you want to override the results database that is configured in the Test Integrations and APIs project with a different results database that has the following details:

- MySQL database as the results database with the URL as `jdbc:mysql://localhost:3306/myresultsdb`
- `com.mysql.cj.jdbc.Driver` as the driver for the MySQL database
- `tester` as the username to access the MySQL database
- `#com.ghc.1!9a8521137E0A12994ACF4434E315A2B9` as the encrypted password to authenticate the user name `tester`

The entry in the **Program Arguments** field must be as follows:

```
-resultDatabaseURL jdbc:mysql://localhost:3306/myresultsdb
-resultDatabaseDriver com.mysql.cj.jdbc.Driver
-resultDatabaseUsername tester
-resultDatabasePassword #com.ghc.1!9a8521137E0A12994ACF4434E315A2B9
```



Note: You must separate entries in the field with white spaces or with a new-line character.

11. Follow the instructions if the API Suite has an environment or secrets configured:
 - a. Click the **ENVIRONMENT** tab, if it is not already open.
 - b. Select the API test environment from the list if there are multiple environments configured in the test asset.
 - c. Select the classification that contains the secrets to be used for the test run.



Notes:

- The test asset that was created in the desktop client and added to the Git repository must have the environments defined as part of the API test project.
- If the test asset contained secrets, then you must create those secrets in the classifications in the project.

The default value for the environment is the environment configured in the test asset. The default value for secrets is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.





Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

a. Click the **VARIABLES** tab, if it is not already open.

b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.



Note: You must have created a file with the variables before you can select the file.


The default value for the variables is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

14. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.

b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

15. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

- a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.



Important: You must have added the following remote hosts to your project that are then displayed under the **Override** column:



- Docker hosts that are registered with Test Hub.
- Kubernetes clusters that are registered with a team space.

**Notes:**

- If the remote Docker hosts or clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote Docker hosts or clusters are added to your project, the added Docker hosts or clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts or clusters are available in your project.
- Select the remote Docker host or cluster from the list when a remote Docker host or cluster is available in your project.
- Select **No override options**, if you selected any remote Docker host or cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of an API Suite.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)


[Test run configurations on page 446](#)

Configuring a run of a Compound Test that contains HTML tests

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains HTML tests to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- When you want to run a Compound Test, you must have authored a Compound Test in HCL DevOps Test UI (Test UI) and must have completed the following tasks when you created the Compound Test:
 - Created a Compound Test under a Web UI project in your workspace.
 - Added a Functional HTML Test under the Compound Test.
 - Used the **startBrowser(Firefox,<URL>)** attribute with the browser value set to *Firefox* in the Functional test script.
-  **Restriction:** Functional test scripts that contain the **startApp()** or **callScript()** attributes are treated as non-HTML scripts and are not supported to be used in test scripts in the HTML tests.
- Linked the dataset that is used in the Functional HTML Test with the Web UI Compound Test only if you want to override this dataset. You must have created the test resources in HCL DevOps Test UI, before committing the test resources to the server project repository. You must have also created the dataset that you want to use to override the dataset linked to the test.
- Ran the Compound Test that contains the HTML tests on Test UI.
- Committed the projects that include the Web UI project (with the Compound Test) and the Functional test (with the HTML tests) to the remote repository.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).

- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.








































The test assets in the selected tab are displayed.

5. Identify the test asset that you want to run by performing any of the following actions:


- Scroll through the list.

 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.


 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:

Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.

- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - a. Click **New filter**.
 - b. Enter a name for the filter.
 - c. Select an operator, and then add a rule or a group of rules.
 - d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.


- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the

repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:



- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the **✕** icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the **✕** icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day

Schedule	Actions
	<ul style="list-style-type: none"> ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p> <div data-bbox="708 632 1421 793" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Repeat every <input type="text" value="1"/> <input type="text" value="Week(s)"/> ▾</p> <p style="text-align: center;"> <input type="radio"/> S <input checked="" type="radio"/> M <input type="radio"/> T <input checked="" type="radio"/> W <input type="radio"/> T <input checked="" type="radio"/> F <input type="radio"/> S </p> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

- a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

- b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.



Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions in this step if the test requires a variable that must be passed to the test at the test run time.

You must configure the supported browser by using a variable if the test has a browser configured, which is different from the one that is supported by Test Hub.


a. Click the **VARIABLES** tab, if it is not already open.

b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.

i. Enter `RTW_WebUI_Browser_Selection` as the name.

ii. Enter `Firefox` as the value, if you want to use Firefox as the browser and override the browser that is specified in the HTML test.

- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.



Note: You must have created a file with the variables before you can select the file.


The default value for the variables is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- Click the **RESULTS** tab.
- Perform the actions listed in the following table to select an option to export the test results:


Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> Select the Jira Xray option. The Select issue field is displayed. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. Select the issue to which you want to export the test results. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> Select the New Jira item option. Continue with the next step.

Export options	Results in...	Actions
None	<p>The test results are not exported.</p> <p> Note: This is the default option.</p>	<p>Perform the following steps when:</p> <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ul style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

 **Important:** You must have added the following remote hosts to your project that are then displayed under the **Override** column:

- Docker hosts that are registered with Test Hub.
- Kubernetes clusters that are registered with a team space.

 **Notes:**

- If the remote Docker hosts or clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote Docker hosts or clusters are added to your project, the added Docker hosts or clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts or clusters are available in your project.
- Select the remote Docker host or cluster from the list when a remote Docker host or cluster is available in your project.
- Select **No override options**, if you selected any remote Docker host or cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Compound Test that contains Functional HTML tests.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Compound Test that contains mobile tests to run by using variables

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains mobile tests to run by using variables from HCL DevOps Test Hub (Test Hub). You must specify the variables for the mobile tests to run on mobile devices that are connected to an agent or a device cloud.

Before you begin

You must have completed the following tasks:

- Created a Compound Test that contains the mobile tests to run on a mobile device in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Created a variable file if you want to import the variables file. The file must contain the details of the UI Test agent or the device cloud to which the mobile device is connected.
- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task

You can run a Compound Test, which contains mobile tests to run on a mobile device, emulator, or simulator that is connected to an agent or a device cloud. You must create the tests in Test UI, and then add the test asset to your project on Test Hub. You can then run the mobile tests on the mobile devices that are connected to any of the following agents or clouds:

- Remote UI Test agents. Requires the variables at test run time as specified in [Variables for the UI Test agent on page 510](#).
- The BitBar Cloud. Requires the variables at test run time as specified in [Variables for the BitBar Cloud on page 511](#).
- The BrowserStack Cloud. Requires the variables at test run time as specified in [Variables for the BrowserStack Cloud on page 511](#).
- The pCloudy Cloud. Requires the variables at test run time as specified in [Variables for the pCloudy Cloud on page 512](#).
- The Perfecto Mobile Cloud. Requires the variables at test run time as specified in [Variables for the Perfecto Mobile Cloud on page 512](#).

You must provide the details of the UI Test agent or the mobile cloud to which the mobile devices are connected as variables in Step 12 on [page 520](#). You can either enter the variables or use the file in which you entered the variables.

You must refer to the following tables for the variables that are required for a successful run:

Table 27. Variables for the UI Test agent

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the UI Test agent. For example, <i>Emulator:Pixel_3a-API_33</i> .
appium.server.host	Specify the hostname or IP address of the UI Test agent.
appium.server.port	Specify the port on the UI Test agent that is configured to communicate with Test Hub.

Table 28. Variables for the BitBar Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the BitBar Cloud. For example, <i>BitBar Cloud:Google Pixel 7 Pro</i> .
bitbar.apikey	Specify the user token generated for your BitBar Cloud account to authenticate your connection with the BitBar Cloud.
bitbar.host	Specify the hostname of the BitBar Cloud instance.
bitbar.project	Specify the name of the project that contains the recorded test.
bitbar.testrun	Specify a name for the test run that must be displayed in the BitBar Cloud dashboard for the test run.

Table 29. Variables for the BrowserStack Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the BrowserStack Cloud. For example, <i>BrowserStack Cloud:Google Pixel 7 Pro</i> .
browserstack.host	Specify the URL of the BrowserStack Cloud instance.
browserstack.username	Specify the user name of your BrowserStack Cloud account.
browserstack.apikey	Specify the API key of your BrowserStack Cloud account to authenticate your connection with the BrowserStack Cloud.
browserstack.project	Specify the name of the project that contains the recorded test.

Table 29. Variables for the BrowserStack Cloud (continued)


Name of the Variable	Action for the Value field
	 Note: This variable is optional.

Table 30. Variables for the pCloudy Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the pCloudy Cloud. For example, <i>pCloudy Cloud:Google Pixel 7 Pro</i> .
pcloudy.apikey	Specify the API key that is generated for your pCloudy Cloud account to authenticate your connection with the pCloudy Cloud.
pcloudy.username	Specify the username with which you can access the pCloudy Cloud instance.
pcloudy.host	Specify the hostname of the pCloudy Cloud instance.

Table 31. Variables for the Perfecto Mobile Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the Perfecto Mobile Cloud. For example, <i>Perfecto Mobile Cloud:Google Pixel 7 Pro</i> .
perfecto.securitytoken	Specify the user token generated for your Perfecto Mobile Cloud account to authenticate your connection with the Perfecto Mobile Cloud.
perfecto.host	Specify the hostname of the Perfecto Mobile Cloud instance.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.


3. Select the branch of the repository that contains the test assets.



















The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:


Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.






















4. Click the **SUITES** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path
- Last Result
- Next Run
- Components

- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

- 6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

- 7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="711 262 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.


10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.


b. Enter the following program argument that must be passed to the test run at run time in the **Program Arguments** field:

Enter `-history testlog` in the **Program Arguments** field.

 **Important:** If the `-history testlog` is not entered in the **Program Arguments** field, the unified report that is generated displays the test steps incorrectly.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.

 **Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.


If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:


The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.

 **Remember:** You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.

 **Note:** If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.





Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Perform the following steps to provide the variables that specify the UI Test agent or cloud to which the mobile device is attached. You can either enter the variables that must be passed to the test at the test run time or import the file that contains the variables.

- a. Click the **VARIABLES** tab, if it is not already open.

- b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.




Note: You must have created a file with the variables before you can select the file.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- a. Click the **RESULTS** tab.

- b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the	Perform the following steps: i. Select the Jira Xray option. The Select issue field is displayed. ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you

Export options	Results in...	Actions
	selected issue that exists in your Xray project in Jira.	<p>can enter the issue key in the Jira Xray project.</p> <p>iii. Select the issue to which you want to export the test results.</p> <p>iv. Continue with the next step.</p>
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	<p>Perform the following steps:</p> <p>i. Select the New Jira item option.</p> <p>ii. Continue with the next step.</p>
None	<p>The test results are not exported.</p> <p> Note: This is the default option.</p>	<p>Perform the following steps when:</p> <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <p>i. Select the None option.</p> <p>ii. Continue with the next step.</p>

14. Perform any of the following actions that depend on whether you are using a remote performance agent or a mobile device cloud:
 - Go to [Step 15 on page 521](#), if you are using an agent.
 - Go to [Step 16 on page 523](#), if you are using a mobile device cloud.
15. Follow the instructions if you want to run the tests on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:
 - a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability and capabilities of the agent are displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
<p>The performance agent that is specified in the test assets is available and no other performance agents are added to the project.</p>	<p>The performance agent that is specified in the test assets is displayed in the Override column.</p>	<p>Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.</p>
<p>The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.</p>	<p>There is no performance agent displayed for override in the Override column.</p>	<p>You cannot run the test on the performance agent.</p> <p>You must wait until the performance agent is available to run the test on the performance agent.</p>
<p>The performance agent that is specified in the test assets is available and there are other performance agents added to the project.</p>	<p>Performance agents that have capabilities that match with the performance agent capabilities specified in the test assets are listed in the Override column as follows:</p> <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. <p>You can find details of both performance agents for the matching capabilities when you hover over the agent in the Override column.</p>	<p>Perform any of the following actions:</p> <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

c. Click **Execute**.

16. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Compound Test that contains mobile tests to run on a mobile device that is connected to an agent or a device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Compound Test that contains mobile tests to run by using a cloud credential

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains mobile tests to run by using a cloud credential from HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Created a Compound Test that contains the mobile tests in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created a cloud credential in your team space that is configured to connect to the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).
- Ensured that you are a member of the cloud credential that is configured to connect to the mobile device cloud. See [Viewing details of cloud credentials on page 439](#).

About this task

You can run a Compound Test, which contains mobile tests to run on mobile devices that are connected to a device cloud. You must create the tests in Test UI, and then add the test asset to your project on Test Hub.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.


4. Click the **SUITES** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.











































Tip: You can hover over the icon in the **Type** column to know the type of the test asset.




Note: You can also identify the type of the asset from the icon that represents the test type as shown in the following table:



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:

- a. Click **New filter**.
- b. Enter a name for the filter.
- c. Select an operator, and then add a rule or a group of rules.
- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.





Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.



8. Select the time for scheduling the test run from the following options:


- No action is required if you want to initiate the test run immediately after you click **Execute**.

 **Important:** Click **Execute** only after you have configured the other settings in this dialog.

 **Note:** The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="708 264 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Perform the following steps in the **ENVIRONMENT** tab:

a. Select the cloud credential of the mobile device cloud from the **Device cloud credentials** list.

The list displays the cloud credentials that are configured in the team space. The name of the cloud credential is appended to the name of the mobile device cloud in the list. For example, if the mobile device cloud provider is *BitBar* and the name of the credential is *BB_Account1*, then the name displayed in the list is *Bit Bar - BB_Account1*.



Note: Access to credentials for mobile device cloud services depends on the status of the provider. If you disable a provider on the **Cloud Credentials** page, you cannot access that device cloud provider from the **Device cloud credentials** list.

If the connection to the mobile device cloud is successful, the **Mobile devices** field is displayed.

- b. Select the devices on which you want to run the tests from the list of devices that are displayed in the **Mobile devices** list.



Note: If you are connected to either the BitBar Cloud or BrowserStack Cloud, the **Projects** field is displayed.

- c. Select the project on the mobile device cloud that you want to use or create a project in the **Projects** field. The project overrides the project that is configured in the test asset.

11. Click **Advanced** to make the following advanced configurations:

- a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

- b. Enter the following program argument that must be passed to the test run at run time in the **Program Arguments** field:

Enter `-history testlog` in the **Program Arguments** field.



Important: If the `-history testlog` is not entered in the **Program Arguments** field, the unified report that is generated displays the test steps incorrectly.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you are running a test asset that contains datasets:

a. Click the **DATA SOURCES** tab, if it is not already open.

b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.




Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.

b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Compound Test that contains mobile tests to run on a mobile device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Test run configurations on page 446](#)

[Management of mobile device cloud credentials on page 432](#)

[Resetting the configuration settings for a test run on page 735](#)

Configuring a run of a Compound Test that contains Web UI tests

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains Web UI tests to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Prerequisites to running tests on page 371](#), if they apply to the test that you want to configure for a run.
- Read and completed the tasks mentioned in [Test run considerations for browser-based Web UI tests on page 376](#), if you want to run tests that use Chrome or Edge browsers.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Created a Compound Test that contains Web UI tests in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).

- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.







The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:


Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.


































4. Click the **SUITES** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path
- Last Result
- Next Run
- Components

e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.





Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.



8. Select the time for scheduling the test run from the following options:


- No action is required if you want to initiate the test run immediately after you click **Execute**.

 **Important:** Click **Execute** only after you have configured the other settings in this dialog.

 **Note:** The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="711 264 1417 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:
 - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.



Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.



If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions in this step if the test requires a variable that must be passed to the test at the test run time.

You must configure the supported browser by using a variable if the test has a browser configured, which is different from the one that is supported by Test Hub.

a. Click the **VARIABLES** tab, if it is not already open.

b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
 - i. Enter `RTW_WebUI_Browser_Selection` as the name.
 - ii. Enter `Firefox` as the value, if you want to use Firefox as the browser and override the browser that is specified in the Web UI test.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.



Note: You must have created a file with the variables before you can select the file.


The default value for the variables is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.

b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	<p>Perform the following steps:</p> <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	<p>Perform the following steps:</p> <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	<p>The test results are not exported.</p> <p> Note: This is the default option.</p>	<p>Perform the following steps when:</p> <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Select any of the following steps that depends on the location to run the test:

- If you want to run the tests on the remote agents, go to [Step 15 on page 542](#).
- If you want to run the tests on a Kubernetes cluster, go to [Step 16 on page 544](#).

15. Follow the instructions if you want to run the tests on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:

- a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability and capabilities of the agent are displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the performance agent.
The performance agent that is specified in the test assets is available and there are other performance agents added to the project.	Performance agents that have capabilities that match with the performance agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the ca- 	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

If...	Then...	Action
	<p>capabilities of the agent in the asset are listed subsequently.</p> <p>You can find details of both performance agents for the matching capabilities when you hover over the agent in the Override column.</p>	

c. Click **Execute**.

16. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.



Important: You must have added the following remote hosts to your project that are then displayed under the **Override** column:

- Docker hosts that are registered with Test Hub.
- Kubernetes clusters that are registered with a team space.



Notes:

- If the remote Docker hosts or clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote Docker hosts or clusters are added to your project, the added Docker hosts or clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts or clusters are available in your project.
- Select the remote Docker host or cluster from the list when a remote Docker host or cluster is available in your project.
- Select **No override options**, if you selected any remote Docker host or cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Compound Test that contains Web UI tests.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Compound Test that contains Web UI tests to run by using variables

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains Web UI tests to run by using variables from HCL DevOps Test Hub (Test Hub). You must specify the variables for the Web UI tests to run on mobile devices that are connected to an agent or a device cloud.

Before you begin

You must have completed the following tasks:

- Created a Compound Test that contains Web UI tests to run on a mobile device in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Created a variable file if you want to import the variables file. The file must contain the details of the UI Test agent or the device cloud to which the mobile device is connected.
- Read and completed the tasks mentioned in [Test run considerations for browser-based Web UI tests on page 376](#), if you want to run tests that use Chrome or Edge browsers.
- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.

- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task

You can run the Compound Test that contains Web UI tests on a mobile device, emulator, or simulator that is connected to an agent or a device cloud. You must create the tests in Test UI, and then add the test asset to your project on Test Hub. You can then run the Compound Test that contains Web UI tests on a mobile device that is connected to any of the following agents or clouds:

- Remote UI Test agents. Requires the variables at test run time as specified in [Variables for the UI Test agent on page 546](#).
- The BitBar Cloud. Requires the variables at test run time as specified in [Variables for the BitBar Cloud on page 547](#).
- The BrowserStack Cloud. Requires the variables at test run time as specified in [Variables for the BrowserStack Cloud on page 547](#).
- The pCloudy Cloud. Requires the variables at test run time as specified in [Variables for the pCloudy Cloud on page 548](#).
- The Perfecto Mobile Cloud. Requires the variables at test run time as specified in [Variables for the Perfecto Mobile Cloud on page 548](#).

You must provide the details of the UI Test agent or the mobile cloud to which the mobile devices are connected as variables in Step [12 on page 556](#). You can either enter the variables or use the file in which you entered the variables.

You must refer to the following tables for the variables that are required for a successful run:

Table 32. Variables for the UI Test agent

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use on the UI Test agent. For example, <i>Chrome(Emulator:Pixel_3a-API_33)</i> .
appium.server.host	Specify the hostname or IP address of the UI Test agent.

Table 32. Variables for the UI Test agent (continued)

Name of the Variable	Action for the Value field
appium.server.port	Specify the port on the UI Test agent that is configured to communicate with Test Hub.

Table 33. Variables for the BitBar Cloud

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use in the BitBar Cloud. For example, <i>Chrome(BitBar Cloud:Google Pixel 7 Pro)</i> .
bitbar.apikey	Specify the user token generated for your BitBar Cloud account to authenticate your connection with the BitBar Cloud.
bitbar.host	Specify the hostname of the BitBar Cloud instance.
bitbar.project	Specify the name of the project that contains the recorded test.
bitbar.testrun	Specify a name for the test run that must be displayed in the BitBar Cloud dashboard for the test run.

Table 34. Variables for the BrowserStack Cloud

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use in the BrowserStack Cloud. For example, <i>Chrome(BrowserStack Cloud:Google Pixel 7 Pro)</i> .
browserstack.host	Specify the URL of the BrowserStack Cloud instance.
browserstack.username	Specify the user name of your BrowserStack Cloud account.

Table 34. Variables for the BrowserStack Cloud (continued)


Name of the Variable	Action for the Value field
browserstack.apikey	Specify the API key of your BrowserStack Cloud account to authenticate your connection with the BrowserStack Cloud.
browserstack.project	Specify the name of the project that contains the recorded test.  Note: This variable is optional.

Table 35. Variables for the pCloudy Cloud

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use in the pCloudy Cloud. For example, <i>Chrome(pCloudy Cloud:Google Pixel 7 Pro)</i> .
pcloudy.apikey	Specify the API key that is generated for your pCloudy Cloud account to authenticate your connection with the pCloudy Cloud.
pcloudy.username	Specify the username with which you can access the pCloudy Cloud instance.
pcloudy.host	Specify the hostname of the pCloudy Cloud instance.

Table 36. Variables for the Perfecto Mobile Cloud

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use in the Perfecto Mobile Cloud. For example, <i>Chrome(Perfecto Mobile Cloud:Google Pixel 7 Pro)</i> .
perfecto.securitytoken	Specify the user token generated for your Perfecto Mobile Cloud account to authenticate your connection with the Perfecto Mobile Cloud.

Table 36. Variables for the Perfecto Mobile Cloud (continued)

Name of the Variable	Action for the Value field
perfecto.host	Specify the hostname of the Perfecto Mobile Cloud instance.

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.




The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:


Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.





































4. Click the **SUITES** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.

- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path
- Last Result
- Next Run
- Components

- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="711 262 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:
 - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.





Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Perform the following steps to provide the variables that specify the UI Test agent or cloud to which the mobile device is attached. You can either enter the variables that must be passed to the test at the test run time or import the file that contains the variables.

- a. Click the **VARIABLES** tab, if it is not already open.

- b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.




Note: You must have created a file with the variables before you can select the file.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- a. Click the **RESULTS** tab.

- b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the	Perform the following steps: i. Select the Jira Xray option. The Select issue field is displayed. ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you

Export options	Results in...	Actions
	selected issue that exists in your Xray project in Jira.	<p>can enter the issue key in the Jira Xray project.</p> <p>iii. Select the issue to which you want to export the test results.</p> <p>iv. Continue with the next step.</p>
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	<p>Perform the following steps:</p> <p>i. Select the New Jira item option.</p> <p>ii. Continue with the next step.</p>
None	<p>The test results are not exported.</p> <p> Note: This is the default option.</p>	<p>Perform the following steps when:</p> <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <p>i. Select the None option.</p> <p>ii. Continue with the next step.</p>

14. Perform any of the following actions that depend on whether you are using a remote performance agent or a mobile device cloud:

- Go to [Step 15 on page 557](#), if you are using an agent.
- Go to [Step 16 on page 559](#), if you are using a mobile device cloud.

15. Follow the instructions if you want to run the tests on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:

- a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability and capabilities of the agent are displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the performance agent.
The performance agent that is specified in the test assets is available and there are other performance agents added to the project.	Performance agents that have capabilities that match with the performance agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. You can find details of both performance agents for the matching capabilities when you hover over the agent in the Override column.	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

c. Click **Execute**.

16. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Compound Test that contains Web UI tests to run on a mobile device that is connected to an agent or a device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Compound Test that contains Web UI tests to run by using a cloud credential

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains Web UI tests to run on devices by using a cloud credential from HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Created a Compound Test that contains the Web UI tests in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).

- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created a cloud credential in your team space that is configured to connect to the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).
- Ensured that you are a member of the cloud credential that is configured to connect to the mobile device cloud. See [Viewing details of cloud credentials on page 439](#).

About this task

You can run a Compound Test, which contains Web UI tests on a device cloud. You must create the tests in Test UI, and then add them to your project on Test Hub.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.


The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:








































Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Click the **SUITES** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:


- Scroll through the list.

 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.


 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:

Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.

- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - a. Click **New filter**.
 - b. Enter a name for the filter.
 - c. Select an operator, and then add a rule or a group of rules.
 - d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.


- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the

repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:



- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day

Schedule	Actions
	<ul style="list-style-type: none"> ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p> <div data-bbox="708 632 1419 793" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Repeat every <input type="text" value="1"/> <input type="text" value="Week(s)"/> ▾</p> <p style="text-align: center;"> <input type="radio"/> S <input checked="" type="radio"/> M <input type="radio"/> T <input checked="" type="radio"/> W <input type="radio"/> T <input checked="" type="radio"/> F <input type="radio"/> S </p> </div> <ul style="list-style-type: none"> e. Set the option to end the recurring test runs from the following options: <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. f. Click Save. <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Perform the following steps in the **ENVIRONMENT** tab:

- a. Select the cloud credential of the mobile device cloud from the **Device cloud credentials** list.

The list displays the cloud credentials that are configured in the team space. The name of the cloud credential is appended to the name of the mobile device cloud in the list. For example, if the mobile device cloud provider is *BitBar* and the name of the credential is *BB_Account1*, then the name displayed in the list is *Bit Bar - BB_Account1*.



Note: Access to credentials for mobile device cloud services depends on the status of the provider. If you disable a provider on the **Cloud Credentials** page, you cannot access that device cloud provider from the **Device cloud credentials** list.

If the connection to the mobile device cloud is successful, the **Mobile devices** field is displayed.

- b. Select the devices on which you want to run the tests from the list of devices that are displayed in the **Mobile devices** list.



Note: If you are connected to either the BitBar Cloud or BrowserStack Cloud, the **Projects** field is displayed.

- c. Select the project on the mobile device cloud that you want to use or create a project in the **Projects** field. The project overrides the project that is configured in the test asset.

11. Click **Advanced** to make the following advanced configurations:

- a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

- b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:
 - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.




Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.

b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Compound Test that contains Web UI tests to run on a device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Test run configurations on page 446](#)

[Management of mobile device cloud credentials on page 432](#)

[Resetting the configuration settings for a test run on page 735](#)

Configuring a run of a Compound Test that contains performance tests

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains performance tests to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Created performance tests under a Compound Test in HCL DevOps Test Performance (Test Performance).
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.













The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:


Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.




























4. Click the **SUITES** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name

- Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.





Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.



8. Select the time for scheduling the test run from the following options:


- No action is required if you want to initiate the test run immediately after you click **Execute**.

 **Important:** Click **Execute** only after you have configured the other settings in this dialog.

 **Note:** The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="708 264 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:
 - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.





Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

a. Click the **VARIABLES** tab, if it is not already open.

b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.



Note: You must have created a file with the variables before you can select the file.

The default value for the variables is null or an empty field.


If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.


b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the	Perform the following steps: i. Select the Jira Xray option. The Select issue field is displayed.

Export options	Results in...	Actions
	selected issue that exists in your Xray project in Jira.	<ul style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ul style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ul style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Select any of the following steps that depends on the location to run the test:
 - If you want to run the tests on the remote agents, go to [Step 15 on page 578](#).
 - If you want to run the tests on a Kubernetes cluster, go to [Step 16 on page 580](#).
15. Follow the instructions if you want to run a test on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:
 - a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the performance agent is displayed.

 **Note:** You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the agent.
The agent that is specified in the test assets is available and other agents added to the project.	Performance agents that have capabilities that match with the agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. You can find details of both performance agents for the matching capabilities when you hover over the performance agent in the Override column.	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

c. Click **Execute**.

The test run is initiated.

16. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.



Important: You must have added the following remote hosts to your project that are then displayed under the **Override** column:

- Docker hosts that are registered with Test Hub.
- Kubernetes clusters that are registered with a team space.



Notes:

- If the remote Docker hosts or clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote Docker hosts or clusters are added to your project, the added Docker hosts or clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts or clusters are available in your project.
- Select the remote Docker host or cluster from the list when a remote Docker host or cluster is available in your project.
- Select **No override options**, if you selected any remote Docker host or cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Compound Test that contains performance tests.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).

- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a mobile test to run by using variables

After you added the test resources that you created in the desktop client to the project, you can configure a mobile test to run by using variables on a mobile device from HCL DevOps Test Hub (Test Hub). You must specify the variables for the mobile test to run on mobile devices that are connected to an agent or a device cloud.

Before you begin

You must have completed the following tasks:

- Created a mobile test in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Created a variable file if you want to import the variables file. The file must contain the details of the UI Test agent or the device cloud to which the mobile device is connected.
- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task

You can run the mobile test after you create the test in Test UI, and then add them to your project on Test Hub. You can then run the mobile test on the mobile devices that are connected to any of the following agents or clouds:

- Remote UI Test agents. Requires the variables at test run time as specified in [Variables for the UI Test agent on page 582](#).
- The BitBar Cloud. Requires the variables at test run time as specified in [Variables for the BitBar Cloud on page 582](#).
- The BrowserStack Cloud. Requires the variables at test run time as specified in [Variables for the BrowserStack Cloud on page 583](#).
- The pCloudy Cloud. Requires the variables at test run time as specified in [Variables for the pCloudy Cloud on page 584](#).
- The Perfecto Mobile Cloud. Requires the variables at test run time as specified in [Variables for the Perfecto Mobile Cloud on page 584](#).

You must provide the details of the UI Test agent or the mobile cloud to which the mobile devices are connected as variables in Step 12 on [page 592](#). You can either enter the variables or use the file in which you entered the variables.

You must refer to the following tables for the variables that are required for a successful run:

Table 37. Variables for the UI Test agent

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the UI Test agent. For example, <i>Emulator:Pixel_3a-API_33</i> .
appium.server.host	Specify the hostname or IP address of the UI Test agent.
appium.server.port	Specify the port on the UI Test agent that is configured to communicate with Test Hub.

Table 38. Variables for the BitBar Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the BitBar Cloud. For example, <i>BitBar Cloud:Google Pixel 7 Pro</i> .

Table 38. Variables for the BitBar Cloud (continued)

Name of the Variable	Action for the Value field
bitbar.apikey	Specify the user token generated for your BitBar Cloud account to authenticate your connection with the BitBar Cloud.
bitbar.host	Specify the hostname of the BitBar Cloud instance.
bitbar.project	Specify the name of the project that contains the recorded test.
bitbar.testrun	Specify a name for the test run that must be displayed in the BitBar Cloud dashboard for the test run.

Table 39. Variables for the BrowserStack Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the BrowserStack Cloud. For example, <i>BrowserStack Cloud:Google Pixel 7 Pro</i> .
browserstack.host	Specify the URL of the BrowserStack Cloud instance.
browserstack.username	Specify the user name of your BrowserStack Cloud account.
browserstack.apikey	Specify the API key of your BrowserStack Cloud account to authenticate your connection with the BrowserStack Cloud.
browserstack.project	Specify the name of the project that contains the recorded test.

Table 39. Variables for the BrowserStack Cloud (continued)


Name of the Variable	Action for the Value field
	 Note: This variable is optional.

Table 40. Variables for the pCloudy Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the pCloudy Cloud. For example, <i>pCloudy Cloud:Google Pixel 7 Pro</i> .
pcloudy.apikey	Specify the API key that is generated for your pCloudy Cloud account to authenticate your connection with the pCloudy Cloud.
pcloudy.username	Specify the username with which you can access the pCloudy Cloud instance.
pcloudy.host	Specify the hostname of the pCloudy Cloud instance.

Table 41. Variables for the Perfecto Mobile Cloud

Name of the Variable	Action for the Value field
Mobile_Device_Selection	Specify the name of the mobile device that is connected to the Perfecto Mobile Cloud. For example, <i>Perfecto Mobile Cloud:Google Pixel 7 Pro</i> .
perfecto.securitytoken	Specify the user token generated for your Perfecto Mobile Cloud account to authenticate your connection with the Perfecto Mobile Cloud.
perfecto.host	Specify the hostname of the Perfecto Mobile Cloud instance.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.


3. Select the branch of the repository that contains the test assets.



















The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:


Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.






















4. Click the **TESTS** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path
- Last Result
- Next Run
- Components

- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

- 6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

- 7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.





Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:

Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="708 264 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input type="text" value="1"/> <input type="text" value="Week(s)"/> <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

Important: The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:
 - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.





Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Perform the following steps to provide the variables that specify the UI Test agent or cloud to which the mobile device is attached. You can either enter the variables that must be passed to the test at the test run time or import the file that contains the variables.

- a. Click the **VARIABLES** tab, if it is not already open.

- b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.




Note: You must have created a file with the variables before you can select the file.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- a. Click the **RESULTS** tab.

- b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the	Perform the following steps: i. Select the Jira Xray option. The Select issue field is displayed. ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you

Export options	Results in...	Actions
	selected issue that exists in your Xray project in Jira.	<p>can enter the issue key in the Jira Xray project.</p> <p>iii. Select the issue to which you want to export the test results.</p> <p>iv. Continue with the next step.</p>
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	<p>Perform the following steps:</p> <p>i. Select the New Jira item option.</p> <p>ii. Continue with the next step.</p>
None	<p>The test results are not exported.</p> <p> Note: This is the default option.</p>	<p>Perform the following steps when:</p> <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <p>i. Select the None option.</p> <p>ii. Continue with the next step.</p>

14. Perform any of the following actions that depend on whether you are using a remote performance agent or a mobile device cloud:
 - Go to [Step 15 on page 593](#), if you are using an agent.
 - Go to [Step 16 on page 595](#), if you are using a mobile device cloud.
15. Follow the instructions if you want to run the tests on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:
 - a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability and capabilities of the agent are displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the performance agent.
The performance agent that is specified in the test assets is available and there are other performance agents added to the project.	Performance agents that have capabilities that match with the performance agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. You can find details of both performance agents for the matching capabilities when you hover over the agent in the Override column.	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

c. Click **Execute**.

16. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a mobile test to be run on a mobile device that is connected to an agent or device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a mobile test to run on a mobile device by using a cloud credential

After you added the test resources that you created in the desktop client to the project, you can configure a mobile test to run on a mobile device cloud by using a cloud credential from HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Created a mobile test in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created a cloud credential in your team space that is configured to connect to the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).
- Ensured that you are a member of the cloud credential that is configured to connect to the mobile device cloud. See [Viewing details of cloud credentials on page 439](#).

About this task

You can run the mobile test on a mobile device cloud. You must create the test in Test UI, and then add it to your project on Test Hub.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.


4. Click the **TESTS** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.











































Tip: You can hover over the icon in the **Type** column to know the type of the test asset.




Note: You can also identify the type of the asset from the icon that represents the test type as shown in the following table:



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:

- a. Click **New filter**.
- b. Enter a name for the filter.
- c. Select an operator, and then add a rule or a group of rules.
- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.





Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.



8. Select the time for scheduling the test run from the following options:


- No action is required if you want to initiate the test run immediately after you click **Execute**.

 **Important:** Click **Execute** only after you have configured the other settings in this dialog.

 **Note:** The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="711 264 1417 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Perform the following steps in the **ENVIRONMENT** tab:

a. Select the cloud credential of the mobile device cloud from the **Device cloud credentials** list.

The list displays the cloud credentials that are configured in the team space. The name of the cloud credential is appended to the name of the mobile device cloud in the list. For example, if the mobile device cloud provider is *BitBar* and the name of the credential is *BB_Account1*, then the name displayed in the list is *Bit Bar - BB_Account1*.



Note: Access to credentials for mobile device cloud services depends on the status of the provider. If you disable a provider on the **Cloud Credentials** page, you cannot access that device cloud provider from the **Device cloud credentials** list.

If the connection to the mobile device cloud is successful, the **Mobile devices** field is displayed.

- b. Select the devices on which you want to run the tests from the list of devices that are displayed in the **Mobile devices** list.



Note: If you are connected to either the BitBar Cloud or BrowserStack Cloud, the **Projects** field is displayed.

- c. Select the project on the mobile device cloud that you want to use or create a project in the **Projects** field. The project overrides the project that is configured in the test asset.

11. Click **Advanced** to make the following advanced configurations:

- a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

- b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.




Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- a. Click the **RESULTS** tab.
- b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Click **Execute**.

The test run is initiated.


Results

You have started or scheduled a test run of a mobile test to run on a mobile device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).

- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Test run configurations on page 446](#)

[Management of mobile device cloud credentials on page 432](#)

[Resetting the configuration settings for a test run on page 735](#)

Configuring a run of a Performance test

After you added the test resources that you created in the desktop client to the project, you can configure a Performance test to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Created a Performance test in HCL DevOps Test Performance (Test Performance).
- Read [Test run considerations for using Machine learning capabilities on page 379](#), if you want to configure a run for either a Performance Test or a Schedule that must be analyzed by using Machine Learning capabilities in Test Hub.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.



















The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:


Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.






















4. Click the **TESTS** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path
- Last Result
- Next Run
- Components

- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.





Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.



8. Select the time for scheduling the test run from the following options:


- No action is required if you want to initiate the test run immediately after you click **Execute**.

 **Important:** Click **Execute** only after you have configured the other settings in this dialog.

 **Note:** The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="708 264 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input type="text" value="1"/> <input type="text" value="Week(s)"/> ▾ <div style="display: flex; justify-content: center; gap: 10px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:
 - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.





Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

a. Click the **VARIABLES** tab, if it is not already open.

b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.



Note: You must have created a file with the variables before you can select the file.

The default value for the variables is null or an empty field.


If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.

b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the	Perform the following steps: i. Select the Jira Xray option. The Select issue field is displayed.

Export options	Results in...	Actions
	selected issue that exists in your Xray project in Jira.	<ul style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ul style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ul style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Select any of the following steps that depends on the location to run the test:

- If you want to run the tests on the remote agents, go to [Step 15 on page 614](#).
- If you want to run the tests on a Kubernetes cluster, go to [Step 16 on page 616](#).

15. Follow the instructions if you want to run a test on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:

a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the performance agent is displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the agent.
The agent that is specified in the test assets is available and other agents added to the project.	Performance agents that have capabilities that match with the agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. You can find details of both performance agents for the matching capabilities when you hover over the performance agent in the Override column.	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

c. Click **Execute**.

The test run is initiated.

16. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.



Important: You must have added the following remote hosts to your project that are then displayed under the **Override** column:

- Docker hosts that are registered with Test Hub.
- Kubernetes clusters that are registered with a team space.



Notes:

- If the remote Docker hosts or clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote Docker hosts or clusters are added to your project, the added Docker hosts or clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts or clusters are available in your project.
- Select the remote Docker host or cluster from the list when a remote Docker host or cluster is available in your project.
- Select **No override options**, if you selected any remote Docker host or cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Performance test.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).

- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Web UI test

After you added the test resources that you created in the desktop client to the project, you can configure a Web UI test to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Prerequisites to running tests on page 371](#), if they apply to the test that you want to configure for a run.
- Read and completed the tasks mentioned in [Test run considerations for browser-based Web UI tests on page 376](#), if you want to run tests that use Chrome or Edge browsers.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Created a Web UI test in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.

3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:

If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.

The test assets in the selected tab are displayed.

5. Identify the test asset that you want to run by performing any of the following actions:


- Scroll through the list.











































Tip: You can hover over the icon in the **Type** column to know the type of the test asset.




Note: You can also identify the type of the asset from the icon that represents the test type as shown in the following table:



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:

- a. Click **New filter**.
- b. Enter a name for the filter.
- c. Select an operator, and then add a rule or a group of rules.
- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.





Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.



8. Select the time for scheduling the test run from the following options:


- No action is required if you want to initiate the test run immediately after you click **Execute**.

 **Important:** Click **Execute** only after you have configured the other settings in this dialog.

 **Note:** The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="711 262 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:
 - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.



Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.



If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions in this step if the test requires a variable that must be passed to the test at the test run time.

You must configure the supported browser by using a variable if the test has a browser configured, which is different from the one that is supported by Test Hub.

a. Click the **VARIABLES** tab, if it is not already open.

b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
 - i. Enter `RTW_WebUI_Browser_Selection` as the name.
 - ii. Enter `Firefox` as the value, if you want to use Firefox as the browser and override the browser that is specified in the Web UI test.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.



Note: You must have created a file with the variables before you can select the file.

The default value for the variables is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Select any of the following steps that depends on the location to run the test:
 - If you want to run the tests on the remote agents, go to [Step 14 on page 625](#).
 - If you want to run the tests on a Kubernetes cluster, go to [Step 16 on page 628](#).
14. Follow the instructions if you want to run the tests on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:

- a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability and capabilities of the agent are displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

- b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the performance agent.
The performance agent that is specified in the test assets is available and there are other performance agents added to the project.	Performance agents that have capabilities that match with the performance agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. 	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

If...	Then...	Action
	<ul style="list-style-type: none"> ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. <p>You can find details of both performance agents for the matching capabilities when you hover over the agent in the Override column.</p>	


c. Click **Execute**.

15. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.

b. Perform the actions listed in the following table to select an option to export the test results:


Export options	Results in...	Actions
Jira Xray	<p>The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.</p>	<p>Perform the following steps:</p> <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	<p>The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.</p>	<p>Perform the following steps:</p> <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.

Export options	Results in...	Actions
None	<p>The test results are not exported.</p> <p> Note: This is the default option.</p>	<p>Perform the following steps when:</p> <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ul style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

16. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

 **Important:** You must have added the following remote hosts to your project that are then displayed under the **Override** column:

- Docker hosts that are registered with Test Hub.
- Kubernetes clusters that are registered with a team space.

 **Notes:**

- If the remote Docker hosts or clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote Docker hosts or clusters are added to your project, the added Docker hosts or clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts or clusters are available in your project.
- Select the remote Docker host or cluster from the list when a remote Docker host or cluster is available in your project.
- Select **No override options**, if you selected any remote Docker host or cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Web UI test.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Web UI test to run by using variables

After you added the test resources that you created in the desktop client to the project, you can configure a Web UI test to run by using variables from HCL DevOps Test Hub (Test Hub). You must specify the variables for the Web UI test to run on mobile devices that are connected to an agent or a device cloud.

Before you begin

You must have completed the following tasks:

- Created a Web UI test to be run on a mobile device in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Created a variable file if you want to import the variables file. The file must contain the details of the UI Test agent or the device cloud to which the mobile device is connected.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task

You can run the Web UI test on a mobile device, emulator, or simulator that is connected to an agent or a device in a device cloud. You must create the test in Test UI, and then add the test asset to your project on Test Hub. You can then run the Web UI test on the devices that are connected to any of the following agents or clouds:

- Remote UI Test agents. Requires the variables at test run time as specified in [Variables for the UI Test agent on page 630](#).
- The BitBar Cloud. Requires the variables at test run time as specified in [Variables for the BitBar Cloud on page 631](#).
- The BrowserStack Cloud. Requires the variables at test run time as specified in [Variables for the BrowserStack Cloud on page 631](#).
- The pCloudy Cloud. Requires the variables at test run time as specified in [Variables for the pCloudy Cloud on page 632](#).
- The Perfecto Mobile Cloud. Requires the variables at test run time as specified in [Variables for the Perfecto Mobile Cloud on page 632](#).

You must provide the details of the UI Test agent or the mobile cloud to which the mobile devices are connected as variables in [Step 12 on page 640](#). You can either enter the variables or use the file in which you entered the variables.

You must refer to the following tables for the variables that are required for a successful run:

Table 42. Variables for the UI Test agent

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use on the UI Test agent. For example, <i>Chrome(Emulator:Pixel_3a-API_33)</i> .
appium.server.host	Specify the hostname or IP address of the UI Test agent.
appium.server.port	Specify the port on the UI Test agent that is configured to communicate with Test Hub.

Table 43. Variables for the BitBar Cloud

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use in the BitBar Cloud. For example, <i>Chrome(BitBar Cloud:Google Pixel 7 Pro)</i> .
bitbar.apikey	Specify the user token generated for your BitBar Cloud account to authenticate your connection with the BitBar Cloud.
bitbar.host	Specify the hostname of the BitBar Cloud instance.
bitbar.project	Specify the name of the project that contains the recorded test.
bitbar.testrun	Specify a name for the test run that must be displayed in the BitBar Cloud dashboard for the test run.

Table 44. Variables for the BrowserStack Cloud

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use in the BrowserStack Cloud. For example, <i>Chrome(BrowserStack Cloud:Google Pixel 7 Pro)</i> .
browserstack.host	Specify the URL of the BrowserStack Cloud instance.
browserstack.username	Specify the user name of your BrowserStack Cloud account.
browserstack.apikey	Specify the API key of your BrowserStack Cloud account to authenticate your connection with the BrowserStack Cloud.
browserstack.project	Specify the name of the project that contains the recorded test.

Table 44. Variables for the BrowserStack Cloud (continued)


Name of the Variable	Action for the Value field
	 Note: This variable is optional.

Table 45. Variables for the pCloudy Cloud

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use in the pCloudy Cloud. For example, <i>Chrome(pCloudy Cloud:Google Pixel 7 Pro)</i> .
pcloudy.apikey	Specify the API key that is generated for your pCloudy Cloud account to authenticate your connection with the pCloudy Cloud.
pcloudy.username	Specify the username with which you can access the pCloudy Cloud instance.
pcloudy.host	Specify the hostname of the pCloudy Cloud instance.

Table 46. Variables for the Perfecto Mobile Cloud

Name of the Variable	Action for the Value field
RTW_WebUI_Browser_Selection	Specify the web browser and the device to use in the Perfecto Mobile Cloud. For example, <i>Chrome(Perfecto Mobile Cloud:Google Pixel 7 Pro)</i> .
perfecto.securitytoken	Specify the user token generated for your Perfecto Mobile Cloud account to authenticate your connection with the Perfecto Mobile Cloud.
perfecto.host	Specify the hostname of the Perfecto Mobile Cloud instance.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.


3. Select the branch of the repository that contains the test assets.



















The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:


Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.






















4. Click the **TESTS** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path
- Last Result
- Next Run
- Components

- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

- 6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

- 7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="711 264 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:
 - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.





Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Perform the following steps to provide the variables that specify the UI Test agent or cloud to which the mobile device is attached. You can either enter the variables that must be passed to the test at the test run time or import the file that contains the variables.

- a. Click the **VARIABLES** tab, if it is not already open.

- b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.




Note: You must have created a file with the variables before you can select the file.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- a. Click the **RESULTS** tab.

- b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the	Perform the following steps: i. Select the Jira Xray option. The Select issue field is displayed. ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you

Export options	Results in...	Actions
	selected issue that exists in your Xray project in Jira.	<p>can enter the issue key in the Jira Xray project.</p> <p>iii. Select the issue to which you want to export the test results.</p> <p>iv. Continue with the next step.</p>
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	<p>Perform the following steps:</p> <p>i. Select the New Jira item option.</p> <p>ii. Continue with the next step.</p>
None	<p>The test results are not exported.</p> <p> Note: This is the default option.</p>	<p>Perform the following steps when:</p> <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <p>i. Select the None option.</p> <p>ii. Continue with the next step.</p>

14. Perform any of the following actions that depend on whether you are using a remote performance agent or a mobile device cloud:
 - Go to [Step 15 on page 641](#), if you are using an agent.
 - Go to [Step 16 on page 643](#), if you are using a mobile device cloud.
15. Follow the instructions if you want to run the tests on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:
 - a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability and capabilities of the agent are displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the performance agent.
The performance agent that is specified in the test assets is available and there are other performance agents added to the project.	Performance agents that have capabilities that match with the performance agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. You can find details of both performance agents for the matching capabilities when you hover over the agent in the Override column.	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

c. Click **Execute**.

16. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Web UI test to run on a device that is connected to an agent or a device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Web UI test to run by using a cloud credential

After you added the test resources that you created in the desktop client to the project, you can configure a Web UI test to run on a device cloud by using a cloud credential from HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Created a Web UI test in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Read [Test run considerations for using mobile device clouds on page 384](#), if you want to configure a run on a mobile device cloud.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).
- Created a cloud credential in your team space that is configured to connect to the mobile device cloud. See [Adding cloud credentials of mobile device clouds on page 435](#).
- Ensured that you are a member of the cloud credential that is configured to connect to the mobile device cloud. See [Viewing details of cloud credentials on page 439](#).

About this task

You can run a Web UI test on a device that is connected to a device cloud. You must create the Web UI test in Test UI, and then add the test asset to your project on Test Hub.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.


4. Click the **TESTS** or the **ADVANCED** tab.
5. Identify the test asset that you want to run by performing any of the following actions:
 - Scroll through the list.











































Tip: You can hover over the icon in the **Type** column to know the type of the test asset.




Note: You can also identify the type of the asset from the icon that represents the test type as shown in the following table:



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:

- a. Click **New filter**.
- b. Enter a name for the filter.
- c. Select an operator, and then add a rule or a group of rules.
- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.





Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.



8. Select the time for scheduling the test run from the following options:


- No action is required if you want to initiate the test run immediately after you click **Execute**.

 **Important:** Click **Execute** only after you have configured the other settings in this dialog.

 **Note:** The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="711 264 1417 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input type="text" value="1"/> <input type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Perform the following steps in the **ENVIRONMENT** tab:

a. Select the cloud credential of the mobile device cloud from the **Device cloud credentials** list.

The list displays the cloud credentials that are configured in the team space. The name of the cloud credential is appended to the name of the mobile device cloud in the list. For example, if the mobile device cloud provider is *BitBar* and the name of the credential is *BB_Account1*, then the name displayed in the list is *Bit Bar - BB_Account1*.



Note: Access to credentials for mobile device cloud services depends on the status of the provider. If you disable a provider on the **Cloud Credentials** page, you cannot access that device cloud provider from the **Device cloud credentials** list.

If the connection to the mobile device cloud is successful, the **Mobile devices** field is displayed.

- b. Select the devices on which you want to run the tests from the list of devices that are displayed in the **Mobile devices** list.



Note: If you are connected to either the BitBar Cloud or BrowserStack Cloud, the **Projects** field is displayed.

- c. Select the project on the mobile device cloud that you want to use or create a project in the **Projects** field. The project overrides the project that is configured in the test asset.

11. Click **Advanced** to make the following advanced configurations:

- a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

- b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.




Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- a. Click the **RESULTS** tab.
- b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Click **Execute**.

The test run is initiated.


Results

You have started or scheduled a test run of a Web UI test to run on a device that is connected to a device cloud.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).

- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Test run configurations on page 446](#)

[Management of mobile device cloud credentials on page 432](#)

[Resetting the configuration settings for a test run on page 735](#)

Configuring a run of a Windows test

After you added the test resources that you created in the desktop client to the project, you can configure a Windows test to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Created a Windows test in HCL DevOps Test UI (Test UI) and added the test asset to the project repository on Test Hub.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task

You can run the Windows tests after you create the tests in Test UI, and then add them to your project on Test Hub.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.


The test assets in the selected tab are displayed.








































5. Identify the test asset that you want to run by performing any of the following actions:


- Scroll through the list.

 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.


 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:

- a. Click **New filter**.
- b. Enter a name for the filter.
- c. Select an operator, and then add a rule or a group of rules.
- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="708 264 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.
- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

- c. Select the dataset that you want to use in the test run from any of the following options:
 - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.



Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Perform the following steps to provide the variables that must be passed to the test at the test run time. You can either enter the variables or import the file that contains the variables.


- a. Click the **VARIABLES** tab, if it is not already open.

- b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.

- i. Enter `windows_Device_Selection` as the name.

- ii. Enter `<IP_address>` as the value, which is the IP address of the computer that contains the UI Test agent.

- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.




Note: You must have created a file with the variables before you can select the file.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- a. Click the **RESULTS** tab.

- b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Select any of the following steps that depends on the location to run the test:

- If you want to run the tests on the remote agents, go to [Step 15 on page 662](#).
- If you want to run the tests on a Kubernetes cluster, go to [Step 16 on page 664](#).

15. Follow the instructions if you want to run a test on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:

- a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the performance agent is displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the agent.
The agent that is specified in the test assets is available and other agents added to the project.	Performance agents that have capabilities that match with the agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. 	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

If...	Then...	Action
	You can find details of both performance agents for the matching capabilities when you hover over the performance agent in the Override column.	

c. Click **Execute**.

The test run is initiated.

16. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.



Important: You must have added the following remote hosts to your project that are then displayed under the **Override** column:

- Docker hosts that are registered with Test Hub.
- Kubernetes clusters that are registered with a team space.



Notes:

- If the remote Docker hosts or clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote Docker hosts or clusters are added to your project, the added Docker hosts or clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts or clusters are available in your project.
- Select the remote Docker host or cluster from the list when a remote Docker host or cluster is available in your project.
- Select **No override options**, if you selected any remote Docker host or cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Windows test.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a JMeter test

After you added the test resources that you created in JMeter to the server project, you can configure a JMeter test to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Test run considerations for JMeter tests on page 376](#).
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that the JMeter extension on Test Hub is enabled before you configure a test run.
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task

Important: You can run the JMeter tests only on the default cluster of Test Hub or a remote Kubernetes cluster. You cannot run the JMeter tests on remote agents or remote Docker hosts that are registered with Test Hub.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.








































The test assets in the selected tab are displayed.

5. Identify the test asset that you want to run by performing any of the following actions:


- Scroll through the list.

 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.


 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:

Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.

- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - a. Click **New filter**.
 - b. Enter a name for the filter.
 - c. Select an operator, and then add a rule or a group of rules.
 - d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.


- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the

repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
<p>A single test run</p>	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
<p>Recurring test runs</p>	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p> <div data-bbox="708 1486 1421 1644" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Repeat every <input style="width: 80px;" type="text" value="1"/> <input style="width: 150px;" type="text" value="Week(s)"/> ▾</p> <p style="text-align: center;"> <input type="radio"/> S <input checked="" type="radio"/> M <input type="radio"/> T <input checked="" type="radio"/> W <input type="radio"/> T <input checked="" type="radio"/> F <input type="radio"/> S </p> </div> Set the option to end the recurring test runs from the following options:

Schedule	Actions
	<ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.


10. Click **Advanced** to make the following advanced configurations:

- a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

- b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

For example, enter **-J** to override any JMeter property that is defined in the properties file.

 **Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.


The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.


b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

12. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

 **Important:** You must have added the registered remote Kubernetes clusters to your project that are then displayed under the **Override** column:

 **Notes:**

- If the remote clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote clusters are added to your project, the added clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote clusters are available in your project.
- Select the remote cluster from the list when a remote cluster is available in your project.
- Select **No override options**, if you selected any remote cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a JMeter test.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a JUnit test

After you added the test resources that contain the Maven project with the JUnit tests to the server project, you can configure a JUnit test run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Test run considerations for JUnit tests on page 378](#) before you configure a JUnit test run.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that the JUnit extension on Test Hub is enabled before you configure a test run. See [Enabling server extensions on page 166](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task



Important: You can run the JUnit tests only on the default cluster of Test Hub or a remote Kubernetes cluster. You cannot run the JUnit tests on remote agents or remote Docker hosts that are registered with Test Hub.

After the test runs, you can view the results of the JUnit test in the following formats:

- Surefire
- JUnit XML Report

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.




The test assets in the selected tab are displayed.


5. Identify the test asset that you want to run by performing any of the following actions:





































- Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.

- d. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path
- Last Result
- Next Run
- Components

- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:



- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.





Important: Click **Execute** only after you have configured the other settings in this dialog.




Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:

Schedule	Actions
A single test run	Perform the following steps: <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start.


Schedule	Actions
	<p>d. Select the Never option in the Repeat every field.</p> <p>e. Click Save.</p> <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p> <div data-bbox="708 1283 1422 1444" data-label="Image"> </div>
	<p>e. Set the option to end the recurring test runs from the following options:</p>

Schedule	Actions
	<ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.


10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

You can enter a variable that points to the `MAVEN_OPTS` property.

b. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, if you have created the JUnit tests by using JDK V12, then you must enter the variable as **JDKOTHER** and the value as **jdkv12**, so that the test run uses the JDK in the path `/data/junit-ext/jdks/jdkv12`.

 **Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.


The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.

b. Perform the actions listed in the following table to select an option to export the test results:


Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

12. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column.

The information about the availability of the default location is displayed.

 **Important:** You must have added the registered remote Kubernetes clusters to your project that are then displayed under the **Override** column:



Notes:

- If the remote clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote clusters are added to your project, the added clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote clusters are available in your project.
- Select the remote cluster from the list when a remote cluster is available in your project.
- Select **No override options**, if you selected any remote cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a JUnit test.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Postman test

After you added the test resources that you created and exported from Postman to the server project, you can configure the Postman tests to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Test run considerations for Postman tests on page 380](#).
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that the Postman extension in Test Hub is enabled before you configure a test run.
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task



Important: You can run the Postman tests only on the default cluster of Test Hub or a remote Kubernetes cluster. You cannot run the Postman tests on remote agents or remote Docker hosts that are registered with Test Hub.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.

Tab	Description
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.










The test assets in the selected tab are displayed.


5. Identify the test asset that you want to run by performing any of the following actions:































- Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.





Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.



8. Select the time for scheduling the test run from the following options:


- No action is required if you want to initiate the test run immediately after you click **Execute**.

 **Important:** Click **Execute** only after you have configured the other settings in this dialog.

 **Note:** The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p>

Schedule	Actions
	<div data-bbox="708 262 1419 422" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Repeat every <input style="width: 50px; text-align: center;" type="text" value="1"/> <input style="width: 100px;" type="text" value="Week(s)"/> ▼ <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> S M T W T F S </div> </div> <p>e. Set the option to end the recurring test runs from the following options:</p> <ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Click **Advanced**, and then enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

You can enter the command options that you use when you run Postman tests by using Newman, as the program arguments.

For example, enter **--verbose**, if you want the details of the test run and each request sent.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.


The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.


b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported. <p> Note: This is the default option.</p>	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

12. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

- a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

 **Important:** You must have added the registered remote Kubernetes clusters to your project that are then displayed under the **Override** column:



Notes:

- If the remote clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote clusters are added to your project, the added clusters are displayed along with their availability status and ownership information.

- b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote clusters are available in your project.
- Select the remote cluster from the list when a remote cluster is available in your project.
- Select **No override options**, if you selected any remote cluster and want to revert to the **Default Cluster** to run the test asset.

- c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Postman test.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Rate Schedule or VU Schedule

After you added the test resources that you created in the desktop client to the project, you can configure a Rate Schedule or VU Schedule to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Created Schedules in HCL DevOps Test Performance (Test Performance) and added the test asset to the project repository on Test Hub.
- Read [Test run considerations for schedules on page 382](#), if you want to configure a run for Rate Schedules or VU Schedules.
- Read [Test run considerations for using Machine learning capabilities on page 379](#), if you want to configure a run for either a Performance Test or a Schedule that must be analyzed by using Machine Learning capabilities in Test Hub.
- Read and completed the tasks mentioned in [Test run considerations for running tests on remote performance agents on page 387](#), if you want to use a remote agent as the location to run the test.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.


The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:




























Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.


4. Identify the test asset that you want to run by performing any of the following actions:













- Scroll through the list.


 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:


Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.


The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

5. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

6. Select the version of the test resources that you want to run by completing any of the following actions:



Note: The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects that are created in any of the desktop products.

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.

- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the **✕** icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the **✕** icon.

7. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:

Schedule	Actions
A single test run	Perform the following steps: <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save.

Schedule	Actions
	The test run is scheduled to run at the selected time on the scheduled date.
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> Minute Hour Day Week Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p> <div data-bbox="708 1178 1422 1339" data-label="Image"> </div> Set the option to end the recurring test runs from the following options: <ul style="list-style-type: none"> Select the Never option, if you do not want the recurring test runs to end. Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. Click Save. <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

8. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.



Important: The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

9. Click **Advanced** to make the following advanced configurations:

- a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

For example, you can set a maximum Java heap size.

- b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

If you want the test log for the test, you must enter `-history testlog` in the **Program Arguments** field.

- c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Follow the instructions if you are running a test asset that contains datasets:

- a. Click the **DATA SOURCES** tab, if it is not already open.

- b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, when you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients and if you have another set of data stored in an Excel or CSV file, then you can select that dataset from the **Override** list.



Remember: You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.



Note: If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.



Note: If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.






Important: If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

- a. Click the **VARIABLES** tab, if it is not already open.
- b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.

 **Note:** You must have created a file with the variables before you can select the file.

The default value for the variables is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.


12. Follow the instructions if you want to override the Resource Monitoring labels in the test asset with the Resource Monitoring labels that you created in Test Hub:

- a. Click the **Resource Monitoring** tab.



Note:

The **Resource Monitoring** tab displays only if you enabled the **Resource Monitoring from Service** option in Test Performance when you created the test asset.

- b. Click , and press the `Ctrl + Space bar` keys, or enter the initial letter of a label to select a label in the list.




Note:

You can select or add labels only if you added the labels in the **Resource Monitoring Sources** page in Test Hub.

13. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- a. Click the **RESULTS** tab.
- b. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. The Select issue field is displayed.

Export options	Results in...	Actions
	The results are exported to the selected issue that exists in your Xray project in Jira.	<ul style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ul style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ul style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

14. Select any of the following steps that depends on the location to run the test:
 - If you want to run the tests on the remote agents, go to [Step 15 on page 701](#).
 - If you want to run the tests on a Kubernetes cluster, go to [Step 16 on page 703](#).
15. Follow the instructions if you want to run a test on the remote performance agent that is connected either to the default Kubernetes cluster or the remote Kubernetes cluster:
 - a. Click the **LOCATION** tab, if it is not already open.

The performance agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the performance agent is displayed.



Note: You must have added performance agents to your project from the **Agents and Intercepts** page for the performance agents to be displayed under the **Override** column.

The default value for the performance agents is null or an empty field if no performance agents were configured in the test asset. If the test asset contains performance agents that are configured, then the default performance agent is the first item to be displayed on the list of performance agents listed in increasing alphabetical order.

b. Select the performance agent where you want to run the test asset in the following scenarios:

If...	Then...	Action
The performance agent that is specified in the test assets is available and no other performance agents are added to the project.	The performance agent that is specified in the test assets is displayed in the Override column.	Select the performance agent in the Host that is also displayed in the Override column as the location to run the test.
The performance agent that is specified in the test assets is not available and no other performance agents are added to the project.	There is no performance agent displayed for override in the Override column.	You cannot run the test on the performance agent. You must wait until the performance agent is available to run the test on the agent.
The agent that is specified in the test assets is available and other agents added to the project.	Performance agents that have capabilities that match with the agent capabilities specified in the test assets are listed in the Override column as follows: <ul style="list-style-type: none"> ▪ Performance agents with capabilities that best match the capabilities of the agent in the asset are at the top of the list. ▪ Performance agents with capabilities that do not match with the capabilities of the agent in the asset are listed subsequently. <p>You can find details of both performance agents for the matching capabilities when you hover over the performance agent in the Override column.</p>	Perform any of the following actions: <ul style="list-style-type: none"> ▪ Select the performance agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select a performance agent in the Override column as the location to run the test.

c. Click **Execute**.

The test run is initiated.

16. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.



Important: You must have added the following remote hosts to your project that are then displayed under the **Override** column:

- Docker hosts that are registered with Test Hub.
- Kubernetes clusters that are registered with a team space.



Notes:

- If the remote Docker hosts or clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote Docker hosts or clusters are added to your project, the added Docker hosts or clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts or clusters are available in your project.
- Select the remote Docker host or cluster from the list when a remote Docker host or cluster is available in your project.
- Select **No override options**, if you selected any remote Docker host or cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Rate Schedule or VU Schedule.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).

- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuring a run of a Selenium test

After you added the test resources that contain the Maven project with the Selenium tests to a server project, you can configure a Selenium test to be run on HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster.

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Test run considerations for Selenium tests on page 383](#) before you configure a Selenium test run.
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that the JUnit extension on Test Hub is enabled before you configure a test run. See [Enabling server extensions on page 166](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

About this task



Important: You can run the Selenium tests only on the default cluster of Test Hub or a remote Kubernetes cluster. You cannot run the Selenium tests on remote agents or remote Docker hosts that are registered with Test Hub.

After the test runs, you can view the results of the Selenium test in the following formats:

- Surefire
- JUnit XML Report

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Select the tab based on the type of test assets that you want to run as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules.	Click the SUITES tab.
You want to run an individual API test, functional test, or performance test.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, and then run Suites and tests.	Click the ADVANCED tab.








































The test assets in the selected tab are displayed.

5. Identify the test asset that you want to run by performing any of the following actions:


- Scroll through the list.

 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.


 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:

Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.

- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests that are in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.


- Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you want to run.

Clicking the test name displays the **Details** panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the

repository. You can also view the history of the test runs for the specific test under the **History** tab of the **Details** panel.

6. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Select the version of the test resources that you want to run by completing any of the following actions:

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

8. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save. <p>The test run is scheduled to run at the selected time on the scheduled date.</p>
Recurring test runs	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> ▪ Minute ▪ Hour ▪ Day ▪ Week ▪ Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p> <div data-bbox="708 1486 1425 1646" data-label="Image"> <p>The image shows a user interface for scheduling recurring test runs. It features a 'Repeat every' field with a dropdown menu set to 'Week(s)'. Below this, there is a row of seven circular buttons representing the days of the week: S (Sunday), M (Monday), T (Tuesday), W (Wednesday), T (Thursday), F (Friday), and S (Saturday). The M, W, and F buttons are highlighted in blue, indicating they are selected for the test runs.</p> </div> Set the option to end the recurring test runs from the following options:

Schedule	Actions
	<ul style="list-style-type: none"> ▪ Select the Never option, if you do not want the recurring test runs to end. ▪ Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. <p>f. Click Save.</p> <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

9. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.


10. Click **Advanced** to make the following advanced configurations:

a. Enter any JVM arguments that must be passed to the test run at run time in the **JVM Arguments** field, if applicable for the test.

You can enter a variable that points to the *MAVEN_OPTS* property.

b. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, if you have created the Selenium tests by using JDK V12, then you must enter the variable as **JDKOTHER** and the value as **jdkv12**, so that the test run uses the JDK in the path `/data/junit-ext/jdks/jdkv12`.

 **Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.


The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

a. Click the **RESULTS** tab.

b. Perform the actions listed in the following table to select an option to export the test results:


Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

12. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column.

The information about the availability of the default location is displayed.

 **Important:** You must have added the registered remote Kubernetes clusters to your project that are then displayed under the **Override** column:



Notes:

- If the remote clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote clusters are added to your project, the added clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote clusters are available in your project.
- Select the remote cluster from the list when a remote cluster is available in your project.
- Select **No override options**, if you selected any remote cluster and want to revert to the **Default Cluster** to run the test asset.

c. Click **Execute**.


The test run is initiated.

Results

You have started or scheduled a test run of a Selenium test.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)

[Test run configurations on page 446](#)

Configuration of test runs in a GitHub Actions Workflow

You can configure test runs of test resources that are in a project repository of HCL DevOps Test Hub (Test Hub) in a GitHub Actions Workflow. You can choose to run test resources that are in your server project in the GitHub Workflow when you do not want to run the test resources without opening the Test Hub project.

Before you configure a test run for any test resources that are in a project repository of Test Hub, you must complete certain tasks. See [Prerequisites to run tests in a GitHub Actions Workflow on page 713](#).

After you complete the prerequisite tasks, you can proceed to configure a test run in a GitHub Workflow. See [Configuring a test run in a GitHub Actions Workflow on page 714](#).

Prerequisites to run tests in a GitHub Actions Workflow

Before you configure a test run for any test resources that are in a project repository of HCL DevOps Test Hub (Test Hub), you must complete certain tasks to run the test resources in a GitHub Actions Workflow.

You must be familiar with creating projects and running test resources from Test Hub.

You must complete the following tasks to set up and use Test Hub:

- Install Test Hub. See [Installation of DevOps Test Hub on page 93](#).
- Possess valid licenses to use Test Hub. See [Management of licenses in DevOps Test Hub on page 179](#).
- Create a project on Test Hub. See [Adding a project on page 1013](#).
- Add a repository that contains test resources to the project repository. See [Adding repositories to a server project on page 1014](#).
- Generate the Offline User Token on Test Hub that is required for connecting to Test Hub from another application such as the GitHub Action. See [Managing access to DevOps Test Hub on page 994](#).
- Open the project to view the details of the test resources that you want to run in a GitHub Actions Workflow. You must make a note of the test resource details from the **Execution** page of Test Hub.

You must note the following details of the test resource that are to be provided in the YAML file for the GitHub Action:

- The URL of the Test Hub instance.
- The name of the team space that contains the project on Test Hub.
- The name of the project on Test Hub that contains the test resources.
- The branch name of the project repository that contains the test resources.
- The ID of the test resource.

You must complete the following tasks outside of Test Hub:

- Create an empty Git repository on any GitHub Enterprise server, if you do not already use Git repositories to store the test resources that you create. For more information about creating Git repositories, refer to [Creating a new repository](#).

If you already use Git repositories to store test resources created, you can use the same Git repositories.



Note: You must have the required privileges in the Git repository to create a GitHub Actions Workflow.

- Enable the GitHub Actions option in the Git repository, if it is not available as an option to use. For more information about enabling the GitHub Actions, refer to [Learn about GitHub Actions](#).

You can continue to configure a run of a test, see [Configuring a test run in a GitHub Actions Workflow on page 714](#).

Configuring a test run in a GitHub Actions Workflow

After you complete the prerequisite tasks to run tests in a GitHub Actions Workflow, you can configure the test run in a GitHub Actions Workflow. The test is in a project on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:


- Completed the tasks that are mentioned in [Prerequisites to run tests in a GitHub Actions Workflow on page 713](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Extended the CA list. See [Extending the trusted CA list on Linux on page 159](#) or [Extending the trusted CA list on Windows systems on page 159](#).

1. Perform the following steps to open the Git repository that you use or created, which has the GitHub Actions option enabled:
 - a. Open the Git repository on your GitHub Enterprise server.
 - b. Create a folder with the name as `.github/workflows` in the root folder of the repository.
 - c. Create an empty YAML file in the `.github/workflows` folder.

The YAML file is used to define the GitHub Action in the Actions Workflow.


2. Perform the following steps to define the YAML file for the GitHub Action:
 - a. Open the [GitHub Marketplace](https://github.com/marketplace?type=) (<https://github.com/marketplace?type=>) in a web browser.
 - b. Search for *HCL DevOps Test Hub GitHub Action*.
 - c. Open the GitHub Action.
 - d. Read through the information provided.
 - e. Copy the contents of the YAML file that is provided in the example by clicking the copy icon

- f. Open the YAML file that is in the `.github/workflows` folder of your repository and paste the copied contents.
- g. Enter the valid details as values for the required attributes in the YAML file by following the actions in the following table:

Attribute	Action
Required attributes	
serverURL	Specify the URL of the Test Hub instance.
offlineToken	Specify the Offline User Token that is generated by Test Hub.
teamSpace	Specify the name of the team space that contains your project on Test Hub.
project	Specify the name of your project on Test Hub.
branch	Specify the name of the branch in the repository that contains the test resources, which can be found in the Details pane of the test resource on the Execution page.
assetId	Specify the ID of the test resource, which can be found under the Git details section in the Details pane of the test resource on the Execution page.
Optional attributes	
environment	Specify the environment that is configured in the test.
datasets	Specify the dataset that must be used at test run time, which replaces the dataset that is configured in a test.
labels	Specify text labels to identify a specific test result when the test run is complete.
secretsCollection	Specify the name of the secret collection that is configured in a test.
variables	Specify the name of the variable and its value if your test requires variables during the test run time.  Note: You can add multiple variables that are separated by a semi-colon.

- h. Save the YAML file.

3. Perform the following steps to set up a Runner for the GitHub Action in GitHub:

- a. Click the Settings icon .
- b. Click **Actions > Runners > New self-hosted runner**.
- c. Follow the on-screen instructions to set up a self-hosted runner on your computer.

After the runner is installed on your computer, you must verify that the runner is ready to run jobs in the GitHub Actions Workflow and is in the *Active* state. The runner is listed in the **Runners** pane in the Git repository.

4. Perform the following steps to start the GitHub Action in the Workflow:

- a. Click the **Actions** option in the Git repository.

The GitHub Action that you configured in the YAML file for Test Hub is displayed as an **Action** under **Workflows**.

- b. Click the *HCL DevOps Test Hub GitHub Action*, and then click **Run workflow**.

Results

You have configured and run a resource that is in your project on Test Hub in a GitHub Action Workflow.

What to do next

You can perform any of the following tasks on the Git repository page by clicking the job that was completed:

- View the status of the test run by expanding the **Execute test** section.
- View the links to the test results and reports listed in the **Execute test** section.

You can perform any of the following tasks from Test Hub:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Configuration of a scan of an application code

You can use HCL DevOps Test Hub (Test Hub) to scan your application code for security vulnerabilities similar to using the HCL® AppScan CodeSweep plugin to scan code in Visual Studio.

Test Hub is enabled to support scanning of your application code that is in a remote repository.

Before you commit your application code to the remote repository, you must complete certain tasks. See [Considerations to scan an application code on page 717](#).

After you commit the assets to the remote repository, you must complete the following tasks:

1. Log in to Test Hub.
2. Open the project in your team space.
3. Add the remote repository to your project.
4. Open the **Execution page**, and then run a scan of the asset.

For detailed instructions, see [Configuring a run to scan the application code on page 719](#).

Considerations to scan an application code

When you want to use HCL DevOps Test Hub (Test Hub) to run a scan of the application code for security vulnerabilities, you must read the considerations that you must take into account and complete certain tasks.

Before you can run a scan of the application code on Test Hub, you must have completed the following tasks:

- Created a file that has the following characteristics:
 - The format of the file must be `yaml`.
 - The name must be `.codesweep`.
- Committed the following assets or resources to the remote repository:
 - The `.codesweep` file.
 - The application code can either be in the root directory or another folder in the root directory.



Note: The default root directory is the directory that contains the `.codesweep` file.

You can specify the following information in the `.codesweep` file:

- The name of the asset that must be displayed as the test asset in the **Execution** page of Test Hub. The name must be specified at the first level in the file as the value of the **name** attribute.
- Optionally, the name of folders that contain the application code.

All files and folders in the root directory are scanned as the default action, if no files or folders are specified under the **folders** attribute.

You can specify the files or folders to scan by using any of the following options:

- Specify the files or folders by using an `XML` file that is in the same format as the HCL AppScan IRX file.
- Specify the files or folders in the `.codesweep` file under the **includes** or **excludes** attributes under the **folders** attribute.

- Optionally, you can specify the name of files or folders that must be included in the scan. You must specify the files or folders as the value of the **includes** attribute under the **folders** attribute.
- Optionally, you can specify the name of files or folders that contain the application code that need not be scanned. You must specify the files or folders as the value of the **excludes** attribute under the **folders** attribute.



Note: You cannot use *wildcard* characters to indicate the files or folders anywhere in the `.codesweep` file.

- Optionally, you can specify the rules that are to be ignored in the scan. You can specify the rules by using any of the following options:
 - Specify a file, which is a copy of `~/appscan/custom_scanners_service_ignore.json` that is created by the CodeSweep plugin for Visual Studio. With this method, you can reuse the preferences that you have configured in your development environment.
 - Specify a list of rules. Each rule is defined by a name and a language identifier.
- Optionally, you can specify the criteria for attributes and their values that specify the verdict of the scan and the range for the security errors in the code.

For example, you can consider the following details:

Where...	Is...
The name of the asset that is to be displayed on the Execution page	<code>mycode</code>
The folder that contains the code to be scanned	<code>use_code</code>
The folder that contains the code that is to be excluded in the scan	<code>do_not_use_code</code>
The value of the attribute <i>Verdict</i>	
The range for the security errors is classified as follows: <ul style="list-style-type: none"> • <code>max-high</code> • <code>max-medium</code> • <code>max-low</code> 	The conditions for a <i>pass</i> verdict are as follows: <ul style="list-style-type: none"> • <code>0</code> • <code>10</code> • <code>100</code>

The contents of the `.codesweep` file is as follows:

```

name: mycode
folders: # default=./, Optional
  configuration: configuration.xml # XML Configuration file, Optional
  includes: # list of folders to include, Optional
    - use_code
  excludes: # list of folders to exclude, Optional
    
```

```

- do_not_use_code
ignore: # Ignore rules, Optional
  file: custom_scanners_service_ignore.json # json file coming from visual code, Optional
  rules: #list of rules to ignore, Optional
    - name: com.ouncelabs.languagelite.nodejs.Rules.NodeJS_PrivateDataExposedviaHTTTPURL # Name,
      Required
      language: NodeJS # Language, Required
    - name: com.ouncelabs.languagelite.nodejs.Rules.NodeJS_PrivateDataExposedviaHTTP # Name,
      Required
      language: NodeJS # Language, Required
verdict: # PASS verdict, Optional
  max-high: 0 # default=0, Optional
  max-medium: 10 # default=0, Optional
  max-low: 100 # default=-1 (no limit), Optional

```

You can continue to configure a run of a scan of the code, see [Configuring a run to scan the application code on page 719](#).



Restriction: You cannot run the test assets on a remote Docker host.

Related information

[Configuration of a scan of an application code on page 716](#)

Configuring a run to scan the application code

After you added the assets in the remote repository to a project in your team space, you can configure a run to scan the application code specified in the `codesweep` file by using the execution service on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Read and completed the tasks mentioned in [Considerations to scan an application code on page 717](#).
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Completed the Xray integration, if you want to export the test results to a Jira issue in your Xray project in Jira. See [Integration with Xray test management for Jira on page 988](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Added the remote Git repository that contains the test resources to the project. See [Adding repositories to a server project on page 1014](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the test assets, and then click **Execution**.
3. Select the branch of the repository that contains the test assets.

The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.

4. Identify the test asset or resource that you want to run by performing any of the following actions:

- Scroll through the list.




























Tip: You can hover over the icon in the **Type** column to know the type of the test asset.




Note: You can also identify the type of the asset from the icon that represents the test type as shown in the following table:

Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test		✓	✓
	Functional test		✓	✓
	Performance test		✓	✓
	AFT Suite	✓		✓
	API Suite	✓		✓




Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset or resource by entering any text contained in the test asset or resource name in the **Search** field.

- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule, or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Test Asset Name
- Test Asset Path

- Last Result
 - Next Run
 - Components
- e. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:




Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

5. Click the **Execute** icon  in the row of the identified test asset.

The **Execute test asset** dialog is displayed.



Notes:

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

6. Select the version of the test resources that you want to run by completing any of the following actions:

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version that is committed, and then followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch of the repository. If you do not select any version, then the latest version is selected for the test run.



Notes:

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✕ icon. As a result, the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you chose for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✕ icon.

7. Select the time for scheduling the test run from the following options:

- No action is required if you want to initiate the test run immediately after you click **Execute**.






Important: Click **Execute** only after you have configured the other settings in this dialog.



Note: The default time for scheduling a run is **Now**.

- Select **Schedule** and perform the following actions if you want to schedule a single test run or configure recurring test runs:


Schedule	Actions
A single test run	Perform the following steps: <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Select the Never option in the Repeat every field. Click Save.

Schedule	Actions
	<p>The test run is scheduled to run at the selected time on the scheduled date.</p>
<p>Recurring test runs</p>	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Calendar icon  in the row of the Start field. Select the date. Select the time at which the test run must start. Set the frequency at which the test runs must run by entering the number in the Repeat every field, and then select the period from the list. You can select from the following options: <ul style="list-style-type: none"> Minute Hour Day Week Month <p>For example, if you want the test run to be run every day at the set time, enter 1 and, then select the Day(s) option.</p> <p>You can also schedule the test to run at the set time on specific weekdays by selecting the Week(s) option. For example, the following image displays the days selected for the test runs as <i>Monday, Wednesday, and Friday</i>:</p> <div data-bbox="708 1178 1421 1339" data-label="Image"> </div> Set the option to end the recurring test runs from the following options: <ul style="list-style-type: none"> Select the Never option, if you do not want the recurring test runs to end. Select the On option, and then click the Calendar icon . You can select the date and time after which the scheduled test runs do not run. Click Save. <p>The recurring test runs are scheduled to start the first run at the selected time on the scheduled date, and to repeat the run at the frequency that you set.</p>

8. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.


 **Important:** The configuration that you set for the test run in the **Execute test asset** dialog is preserved when you run the same test again. Those changes are not visible when another user logs in to Test Hub. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

9. Follow the instructions if you want to export the test results to a Jira issue in your Xray project in Jira:

- a. Click the **RESULTS** tab.
- b. Perform the actions listed in the following table to select an option to export the test results:


Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> i. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> ii. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. iii. Select the issue to which you want to export the test results. iv. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> i. Select the New Jira item option. ii. Continue with the next step.

Export options	Results in...	Actions
None	<p>The test results are not exported.</p> <p> Note: This is the default option.</p>	<p>Perform the following steps when:</p> <ul style="list-style-type: none"> ▪ You do not want to export the test results. ▪ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> i. Select the None option. ii. Continue with the next step.

10. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

- a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

 **Important:** You must have added the registered remote Kubernetes clusters to your project that are then displayed under the **Override** column:

 **Notes:**

- If the remote clusters are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Test Hub.
- If remote clusters are added to your project, the added clusters are displayed along with their availability status and ownership information.

- b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote clusters are available in your project.
- Select the remote cluster from the list when a remote cluster is available in your project.
- Select **No override options**, if you selected any remote cluster and want to revert to the **Default Cluster** to run the test asset.

- c. Click **Execute**.


The test run is initiated.

Results

You have configured and either started or scheduled a run of a scan of the application code that is specified in the `codesweep` file.

What to do next

You can perform any of the following tasks:

- Stop the test run at any point after the test run is initiated from the **Execution** page. See [Stopping test runs on page 736](#).
- Cancel a scheduled test run from the **Execution** page. See [Canceling scheduled test runs on page 739](#).
- View all the states of the test asset by clicking the **Show in the Progress page** icon  for the test asset for which you started or scheduled the run. See [Viewing the state of test assets on page 727](#).
- View the progress of the test from the **Progress** page. See [Viewing the progress of running test assets on page 731](#).
- View the results, reports, and logs of the test from the **Results** page after the test completes the run. See [Test results on page 788](#).

Related information

[Resetting the configuration settings for a test run on page 735](#)


[Configuration of a scan of an application code on page 716](#)

[Test run configurations on page 446](#)

Management of running tests

Find information about the tasks that you can perform on a test that you configured for a run either while it runs or after it completes the run.


Viewing the state of test assets

After you initiate or schedule a test run on the **Execution** page, you can view the state of the test assets on the **Progress** page by clicking the **Show in the Progress page** icon  that is enabled for that test asset on the **Execution** page. You can view all the states of the test asset from the past hour such as `In transition`, `Running`, `Scheduled`, `Completed`, or `Stopped`.

Before you begin

You must have initiated runs of the test assets in your project from the **Execution** page.



Note: The **Show in the Progress page** icon  is enabled for the test asset only if a run of the test asset is either started or scheduled.

1. Click **Execution**, if you are not already on the **Execution** page.
2. Select the branch of the repository that contains the test assets.


The test assets that are contained in the selected branch of the repository are displayed in the following tabs on the **Execution** page:




























Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.


3. Identify the test asset or resource that you want to run by performing any of the following actions:













- Scroll through the list.

 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.


 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:

Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			




Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset or resource by entering any text contained in the test asset or resource name in the **Search** field.

- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all functional tests in the selected branch of the repository.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then add a rule, or a group of rules.
 - Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- a. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.

- b. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

4. Click the **Show in the Progress page** icon  in the **Actions** column of the test asset.

The **Progress** page is displayed with the following details:

- The different states of the test asset runs that were initiated or completed in the past hour are displayed as follows:
 - In transition
 - Running
 - Completed
 - Stopped by User
 - Canceled
- The ID of the selected test asset is applied as a filter and is displayed in the filter row.

The applied filter is removed along with the displayed items for this filter when you perform any of the following actions on the **Progress** page:


- You click **Hide inactive**, and then click **Show all**.
- You visit any other page in the Navigation pane and return to the **Progress** page.

Results

You have viewed all the states of a specific test asset runs that were initiated during the past hour.

What to do next

You can choose to perform any of the following tasks from the **Progress** page:

- Stop test assets that are running. See [Stopping test runs on page 736](#).
- Cancel scheduled runs. See [Canceling scheduled test runs on page 739](#).
- View the results or the execution log of the test asset from the **Actions** column by clicking the **Open action menu** icon . See [Checking logs on page 734](#).
- View the results, reports, and logs of the completed test run from the **Results** page. See [Test results on page 788](#).

Viewing the progress of running test assets

After you initiate or schedule a test run on the **Execution** page, you can view the progress of the test assets that are running or scheduled to run, from the **Progress** page. You can also view the test assets that have completed their run, are stopped or canceled, during the past hour.

Before you begin

You must have initiated runs of the test assets in your project from the **Execution** page.

1. Open your project and click **Progress**.

Any one of the following views is displayed:

- No test assets are displayed in the **Progress** page if there are no test assets that are running, scheduled to run, or have completed their run in the past hour.
- Test assets in your project that are in any of the following states are listed:
 - Running
 - Scheduled
 - Completed
 - Stopped
 - Canceled
- No test assets are displayed but the option **Show all** is enabled. You can click **Show all** to display the test assets that ran in the past hour. Test assets that were hidden from the display when **Hide inactive** was clicked are displayed.



Note: Test assets are automatically removed from the display after *60 minutes* of being added to the **Progress** page. You can hide the test assets from the display by clicking **Hide inactive**, so that you can view only the test assets that were added to the **Progress** page in the past hour. You cannot use the **Progress** page to view the history of test assets that were run from the **Execution** page.

2. Identify the test asset by performing any of the following actions:

- Identify the test asset or resource by scrolling through the list.
- Search for the test asset by entering any text contained in the test asset name in the **Search** text box.
- Create a filter query by using the **New filter** option by performing the following steps:
 - a. Click **New filter**.
 - b. Select an operator, and then add a rule, or a group of rules.
 - c. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
- Name
- Started by

- Started on
 - Finished on
 - Status
 - Verdict
- d. Apply the filter query to filter the assets based on the query.


The test assets that match the filter criteria are displayed.

- e. Save the filter query by performing the following steps, if you want to reuse the filter query later:

- i. Click **Save**.
 - ii. Enter a name for the filter query.
 - iii. Click **Save**.
- Retrieve a saved filter, if you have saved filter queries earlier by performing the following steps:



Note: To open the filter query, you must have created and saved a filter query.




- a. Click the **Open filters** icon .
- b. Select the saved filter in the **Filters** dialog.
- c. Click **Apply** to apply the filter.

The test assets that match the filter criteria are displayed.

- Sort the test assets by the user who started the test by clicking the **Started by** column header, and find the test asset listed against your name.
- Sort the test assets by the type of the test assets by clicking the **Type** column header, and find the test asset type you ran or scheduled.
- Click **Hide inactive** to only display the test assets that are running or scheduled to run. This action removes all completed, canceled, stopped, or failed test assets from the display.



Note: The items in the rows are sorted in an ascending or descending order when the column header is clicked. The following icons show the status of sorting in a column:

-  items in the column are not sorted.
-  items in the column are sorted in the ascending order or with the earliest record on top.
-  items in the column are sorted in the descending order or with the latest record on top.

You can view the following details of the test assets on the **Progress** page:

Column header	Description
Type	Displays the icon for the type of the test asset.

Column header	Description
Name	<p>Displays the name of the test asset added to the project from the Git repository.</p> <p>Clicking the test name displays the Details panel. You can view the details of the test such as the description, the branch in the repository that contains the asset, the Git details, and the details of the commits to the repository. You can also view the history of the test runs for the specific test under the History tab of the Details panel.</p>
Started by	Displays the name of the user who started the run of the test asset.
Started on	Displays the date and time when the test run was started.
Finished on	Displays the date and time when the test run completed the run.
Status	<p>Displays the state of the test asset progressively.</p> <p>For example, after the run is in the <code>Initiated</code> state, it moves on to the <code>Running</code> state and ends in its final state depending on the verdict of the test run as either <code>Completed</code>, <code>Failed</code>, or <code>Inconclusive</code>.</p> <p>If the test run is stopped or a scheduled run is canceled, the status displayed is <code>Stopped by User</code> for a stopped run or <code>Canceled</code> for a canceled test.</p>

From the **Actions** column, you can perform the following operations on your test asset:

- Stop a run.
- Cancel a scheduled run.
- Cancel scheduled recurring test runs.
- View the test log.
- View the result of the test asset on the **Results** page.
- Query the test result data.


Results

You have viewed the progress of the test assets that you ran from the **Execution** page.

What to do next

You can choose to perform any of the following tasks from the **Progress** page:

- Stop test assets that are running. See [Stopping test runs on page 736](#).
- Cancel scheduled runs. See [Canceling scheduled test runs on page 739](#).

- View the results or the execution log of the test asset from the **Actions** column by clicking the **Open action menu** icon . See [Checking logs on page 734](#).
- View the results, reports, and logs of the completed test run from the **Results** page. See [Test results on page 788](#).
- Query the test results data. See [Test log queries on page 823](#).

Checking logs

To verify how the test ran or to debug test run failure, you can check the `Test Log` and `Execution log`.

About this task

The `Test Log` displays the interaction between HCL DevOps Test Hub (Test Hub) and the application or system under test. After the test run completes, the verdict of the run can be *Pass*, *Fail*, or *Inconclusive*. If the verdict of the run is *Pass*, the `Test Log` is available in the **Results** page.



The screenshot shows a test run summary for 'cmptest-1'. The test is in a 'Complete' state with a 'Pass' verdict. The interface is divided into 'DETAILS' and 'Reports' sections.

DETAILS		Reports
Git Repo	git@github.com:cmptest-1:master /cmptest-server-sample.git	Statistics Report
Git Branch	master	Test Log
Git Path	WebUIProjectCustomCode/CompoundTests/cmptest-1.testsuite	Mobile and Web UI Report

The `Execution log` displays the console messages of the run time process that runs the test. This log is useful in determining the cause of the failure if the verdict of the run is *Fail* or *Inconclusive*. You can view the `Execution log` from the **Progress** page.

Viewing the `Test Log`

1. Go to the **Results** page and identify the test that you ran.
2. Click the test so that the **Reports** panel is displayed.
3. Click the **Test Log** in the **Reports** panel.

The `Test Log` is displayed in a browser window.

Viewing the `Execution log`

4. Go to the **Progress** page and identify the test that is in the *Running* state.
5. Click the **Open action menu** icon.
6. Click **Execution log**.

The `Execution log` is displayed in a browser window.

Related information

[Test run configurations on page 446](#)

Resetting the configuration settings for a test run

When you are configuring a test run either as a first-run or when repeating the run, and you do not want to proceed with the settings configured or saved for the test run, you can reset the configuration settings. Resetting the configuration reverts all the settings to their default values for the test run.

Before you begin

You must be a member of the project with the *Owner* or *Tester* role to run the tests.

You must ensure that you have the **Execute test asset** dialog is displayed before you proceed with resetting the configuration settings for the test run.

About this task

You can reset the configuration settings when you are configuring a test run either for its first run or when you are repeating the test run.

1. Click **Reset** in the **Execute test asset** dialog at any point when you want to reset the configuration settings for the test run.



Notes:

- If you have configured some or all of the settings for the current test run and you do not want to continue with those settings, clicking **Reset** resets the settings to their default values.
- If you are repeating a test run and do not want to use the saved settings from a previous run, clicking **Reset** reverts all the saved settings to their default values.

The settings in the **Execute test asset** dialog are reset to their default values.

2. Use the following table to find the default values for each of the settings in the **Execute test asset** dialog:

Window or Tab	Field or Setting	Default value is
Execute test asset	Scheduling the test run	Now is selected.
Execute test asset	Label for settings	Null or empty field.
Advanced	JVM Arguments	Null or empty field.
	Program Arguments	Null or empty field.
	Environment Variables	Null or empty field.

Window or Tab	Field or Setting	Default value is
ENVIRONMENT tab	API test environment	The environment configured in the test asset.
	Secrets collection	Null or empty field.
DATA SOURCES tab	Override list	Null or empty field if the asset has no dataset.
		The dataset in the test asset if the asset has one dataset.
		The first dataset on the list of datasets, which are listed in increasing alphabetical order, if the asset has multiple datasets.
VARIABLES tab		Null or empty field.
LOCATION tab	Agents	Null or empty field if the asset has no agents configured.
		The first agent on the list of agents, which are listed in increasing alphabetical order, if the asset has multiple agents configured.
	Docker host	Internal Docker host if no remote Docker hosts or clusters are added.
		No override options if other Docker hosts or clusters are saved in previous runs.

What to do next

You can complete configuring the settings that you want for the test run. See [Test run configurations on page 446](#).

Stopping test runs

You might want to stop a test run or multiple tests that are running when you realized that you did not configure all the settings or you want to change a few settings for the test runs.

Before you begin


You must have initiated a run for one or multiple test assets from the **Execution** page.

About this task



You can stop either a running test or multiple running tests from the **Progress** page. You can stop the test if it is in either the `In transition` or `Running` state. You cannot stop the test run if the test has already completed its run.

1. Go to the **Progress** page and identify the tests in any of the following ways that you want to stop:

You can select the tests that satisfy the following conditions:

- The test must be either in the *In transition* OR *Running* state.
- The **Stop execution** icon  in the **Actions** column must be enabled for the test.
- The checkbox must be enabled for the test.


2. Perform the actions in the following table:

If...	Action...
You want to stop any test that is running	Perform any of the following actions: <ul style="list-style-type: none"> ◦ Click the Stop execution icon  in the Actions column of the selected test. ◦ Click the checkbox in the row of the selected test, and then click Stop selected. The Stop execution dialog is displayed.
You want to stop multiple tests that are running	Perform any of the following actions: <ul style="list-style-type: none"> ◦ Click the checkbox in the row of the selected tests, and then click Stop selected. ◦ Click the Select all checkbox in the header row, and then click Stop selected. <div style="text-align: center;">  Note: All running tests are selected when you click the Select all checkbox. </div> The Stop execution dialog is displayed.

3. Perform the following steps in the **Stop execution** dialog that is displayed:

- Set the timeout period for stopping the test run. Enter a numeric value and select the unit from the options available as *Seconds*, *Minutes*, or *Hours*.

The time out period is the time during which the test run is allowed to stop on its own and after the timeout period, the test is forced to stop abruptly.

 **Important:** If the timeout period is not set, a value of *30 seconds* is considered as the default timeout period.

- Perform any of the following actions:

- Keep the **Capture results** option selected, if you want the results to be captured for the test.

The captured results for the test are available to view from the **Results** page.

- Clear this option if you do not want to capture the results.



Note: The **Capture results** option is selected as the default option.



Important: The results are always captured for API suites even if you clear the **Capture results** option.

c. Perform any of the following actions:

- Keep the **Execute 'finally-block', if present** option selected, if you want the *finally-block* code in the test script, if present, to be run before the test run is stopped.
- Clear this option if you do not want to run the *finally-block* code in the test script.



Note: The option **Execute 'finally-block', if present** is selected as the default option.



Restriction: The *tear-down* steps equivalent to the **'finally-block'** code, if present in API suites are not controllable at test runtime. The **Execute 'finally-block', if present** option has no impact on API suites.

d. Click **Stop execution**.

A notification is displayed that the running tests are stopped.




Note: If the test run completes before you can configure the options for stopping the test run, a notification is displayed in the **Stop execution** dialog that the test asset has completed its run.

Results

You have successfully stopped a running test or stopped multiple running tests. The stopped test runs are displayed on the **Progress** page with the status as *Stopped by User*. Stopped tests are not considered in the count of tests executed that are displayed on the **Overview** page.

What to do next

You can re-initiate the test run from the **Execution** page or from the **Results** page by completing the configurations that you want for the test.

You can view the results or the execution log of the stopped test from the **Actions** column on the **Progress** page by clicking the **Open action menu** icon .

Related information

[Test run configurations on page 446](#)

[Viewing the progress of running test assets on page 731](#)

[Canceling scheduled test runs on page 739](#)

Canceling scheduled test runs

You might want to cancel a scheduled test run, scheduled recurring test runs, or multiple scheduled test runs when you realized that you did not configure all the settings for the runs, want to change a few settings, or do not want those tests to run.

Before you begin

You must have scheduled runs for one or multiple test assets from the **Execution** page.

About this task

You can cancel a scheduled test run, all instances of a scheduled recurring test runs, or multiple scheduled tests from the **Progress** page.

1. Perform any of the following actions:


- Click **Scheduled** under the **Next Run** column on the **Execution** page.

The test is displayed on the **Progress** page. If you selected a test that was scheduled as a recurring test, then all the tests that ran and the next scheduled test run are displayed.

Go to [Step 2 on page 739](#).

- Go to the **Progress** page and identify the tests in any of the following ways that you want to cancel:



You can select the tests that satisfy the following conditions:

- The test must be in the `Scheduled` state.
- The **Cancel** icon  in the **Actions** column must be enabled for the test.
- The checkbox must be enabled for the test.

Go to [Step 2 on page 739](#).

2. Perform the actions in the following table:

If...	Action...
You want to cancel any test run that is scheduled.	Perform any of the following actions:

If...	Action...
You want to cancel all scheduled instances of the recurring test runs.	<ul style="list-style-type: none"> ◦ Click the Cancel icon  in the Actions column of the test that displays <i>Scheduled</i> as the status. ◦ Click the checkbox in the row of the test that is scheduled, and then click Stop selected. <p>The Cancel scheduled execution dialog is displayed.</p> <p>Go to Step 3 on page 740.</p>
You want to stop multiple test runs that are scheduled	<p>Perform any of the following actions:</p> <ul style="list-style-type: none"> ◦ Click the checkbox in the row of the selected tests, and then click Stop selected. ◦ Click the Select all checkbox in the header row, and then click Stop selected. <p> Note: All scheduled tests are selected when you click the Select all checkbox.</p> <p>The Cancel scheduled execution dialog is displayed.</p> <p>Go to Step 3 on page 740.</p>

3. Click **Yes** in the **Cancel scheduled execution** dialog.

Result

.

A notification is displayed that the scheduled test runs are canceled.

Results

You have successfully canceled a scheduled test, all instances of the scheduled recurring tests, or multiple scheduled tests. The canceled test runs are displayed on the **Progress** page with the status as *Canceled*. The canceled test runs are not displayed on the **Results** page and are not considered in the count of tests executed that are displayed on the **Overview** page.

What to do next

You can re-initiate the test run from the **Execution** page by completing the configurations that you want for the test.

Related information

[Test run configurations on page 446](#)

[Viewing the progress of running test assets on page 731](#)

[Stopping test runs on page 736](#)

Management of virtualized services

You can find information about the tasks that you can perform on and manage the virtualized services that run on HCL DevOps Test Hub (Test Hub). You can start or stop the virtual services that are available to Test Hub. You can view the routing rules and usage statistics of virtualized services, agents, or intercepts that are connected to Test Hub.

You can create stubs or virtual services in HCL DevOps Test Integrations and APIs (Test Integrations and APIs). You must commit the virtual service resources to a remote Git repository, and then add the Git repository to your project on Test Hub if you used Test Integrations and APIs to create virtual service resources.

You can start a running instance of the virtual services from Test Hub to run on the following locations:

- A registered API agent.
- The default Kubernetes cluster of Test Hub.
- A remote Kubernetes cluster.

You can run all virtual services on registered API agents except the virtual services that use CICS transport, DB2/Z transport, or Istio.

The default Kubernetes cluster or a remote Kubernetes cluster supports running of the following types of virtual services:

- Virtual services that utilize the WebSphere® MQ transport.
- Virtual services that utilize the HTTP transport.



Restriction: You cannot run virtual services on a remote Docker host.

If running a virtual service in a Kubernetes cluster, the following are the considerations that decide the selection of a container:

- Before a container is selected to run a virtual service, it is checked for a previous failure to start a virtual service or for a low heap size, and other conditions. If the container was not found suitable, then it is not selected for running the virtual service.
- Multiple virtual services can be run in the same container if the following conditions are satisfied:
 - Subsequent virtual services are started by the same project member.
 - All virtual services are created and contained in the same Test Integrations and APIs project.

- The requests to run the virtual services refer to the same version, meaning that the git commit IDs on which they are based are the same. If you use the UI, this means that the **Version** field in the **Execute virtual service** dialog are the same.
- The settings that you configure in the **Advanced** settings for the virtual services are the same.
- The datasets selected for an override are the same, if datasets are used.

Working with virtual services

You can perform the following tasks on virtual services on Test Hub:


- Read about the considerations before you configure a run of the virtual services. See [Prerequisites for running HTTP virtual services on page 742](#).
- Read about the considerations before you configure a run of the virtual service on an API agent. See [Considerations for running virtual services on remote API agents on page 395](#).
- Set up the HTTP proxy, if you have configured stubs in the test assets to use an HTTP proxy to route requests. See [Setting up intercepts on page 744](#).
- View all intercepts that are registered with Test Hub including those that you have not configured. See [Viewing intercepts that are registered with a team space on DevOps Test Hub on page 746](#).
- View virtual service resources in the test assets that are in the repositories added to a project on Test Hub. See [Viewing virtual service resources on page 747](#).
- Configure runs of virtual services. See [Configuring a run of a virtual service on page 750](#).
- Configure runs of HTTP virtual services to run without using proxies. See [Running HTTP virtual services in Kubernetes cluster without using proxies on page 765](#).
- View running instances of virtual services. See [Viewing running instances of virtual services on page 770](#).
- View configurations of a running virtual service instance. See [Viewing configurations of running instances of virtual services on page 775](#).
- Modify the configurations of a running virtual service instance. See [Modifying configurations of running instances of virtual services on page 780](#).
- View details of the usage statistics of the virtual services. See [Viewing usage statistics of virtual services on page 784](#).
- View routing rules of the intercepts. See [Viewing routing rules of the virtual services on page 783](#).
- Stop a running virtual service instance. See [Stopping virtual services on page 787](#).

Prerequisites for running HTTP virtual services

You can find information about the prerequisite tasks that you must complete before you configure a run of the HTTP virtual services in HCL DevOps Test Hub (Test Hub).


Prerequisites for running HTTP virtual services in a Kubernetes cluster

You can find information about the tasks that you must complete before you configure a run of the HTTP virtual services in the HCL DevOps Test Hub (Test Hub) cluster or a remote Kubernetes cluster.

 **Important:** The prerequisites do not apply when you use an API agent to run the HTTP virtual services.

Conditions for running of HTTP virtual services

Before you run the HTTP virtual services in the Test Hub cluster or a remote Kubernetes cluster, you must check for the following conditions and perform the actions indicated:

If...	Then...
You want to route the HTTP traffic via the HTTP proxy either to the virtual service or live systems, based on routing rules.	You must set up the HTTP proxy. See Setting up intercepts on page 744 .
You want to route the HTTP traffic to the virtual service via the HTTP proxy and the proxy is not configured to trust all certificates.	The trust store must be configured to include the certificate authority for the server certificate. See Certificate authority: Importing and extending lists on page 156 .
A client application is sending requests directly to the virtual service.	The client application must trust the certificate authority for the server certificate. See Certificate authority: Importing and extending lists on page 156 .
You are working with Test Hub V10.1.3, which is installed on Ubuntu.	<p>HTTP virtual services are exposed via hostnames are of the following form:</p> <pre data-bbox="829 1171 1419 1213">in-<code><unique_id></code>.<code><INGRESS_DOMAIN></code></pre> <p> Note: The DNS used by a computer that runs a client application or the HTTP proxy that routes traffic to the virtual service needs to resolve such hostnames to the IP address of the ingress domain.</p> <p>You must perform the following actions that depend on the method you have used to set up the ingress domain:</p> <ul data-bbox="878 1619 1393 1770" style="list-style-type: none"> • If you used <code>nip.io</code> to form the ingress domain, then that virtual services host names are automatically resolved to the IP address of the ingress domain. <p>For example, if you used <code>nip.io</code> to form the ingress domain as follows:</p>

If...	Then...
	<p data-bbox="915 262 1110 289"><code>10.1.2.3.nip.io</code></p> <p data-bbox="896 310 1398 384">The virtual services hostnames are resolved by <code>nip.io</code> to the IP address as <code>10.1.2.3</code>.</p> <ul data-bbox="880 394 1421 625" style="list-style-type: none"> • If your ingress domain does not use <code>nip.io</code> then you must ensure that the DNS in use by the client application or the HTTP proxy can resolve hostnames of the virtual services to the IP address of the ingress domain. For example, if the ingress domain is as follows: <p data-bbox="915 646 1104 674"><code>myhost.mycom.com</code></p> <p data-bbox="896 695 1414 768">The DNS must resolve <code>*.myhost.mycom.com</code> to the IP address of <code>myhost.mycom.com</code>.</p>

HTTPS virtual service endpoints

When you want to run an HTTP virtual service where the HTTP transport settings have **Use SSL** enabled, the virtual service endpoint that is exposed for the virtual service instance expects an HTTPS request on port 443. The exposed endpoint is managed by the same ingress controller or the HAProxy that is used for the server endpoints. The SSL/TLS termination occurs at the endpoint before the request is routed to the virtual service instance and results in the following implications:

- The **Virtualization (Server) SSL** options that are specified in the HTTP transport have no effect.
- The certificate served by the exposed virtual service endpoint is the same as that of server that is signed by the server certificate authority.

Related information

[Setting up intercepts on page 744](#)

[Running HTTP virtual services in Kubernetes cluster without using proxies on page 765](#)

Setting up intercepts

When you want to run virtual services on HCL DevOps Test Hub (Test Hub), you can set up intercepts to route requests to the virtual service. You must configure the intercepts with the Test Hub URL, team space ID, and offline user token that is generated from Test Hub so that the intercepts can register with Test Hub.

Examples of intercepts include the HTTP proxy component of HCL DevOps Test Virtualization Control Panel (Test Virtualization Control Panel) and the HCL DevOps Test Integrations and APIs (Test Integrations and APIs) Java virtualization agent available from the Test Virtualization Control Panel tools page.

Before you begin

You must have completed the following tasks:

- Installed the intercept from Test Virtualization Control Panel of the same version as that of Test Hub. For example, installed the HTTP proxy.
- Read the following procedures:
 - Generating offline user tokens. See [Managing access to DevOps Test Hub on page 994](#).
 - Viewing the *Team Space ID* as the URL alias of the team space. See [Viewing the configuration of a team space on page 1003](#).


About this task

You must edit the `registration.xml` file for the intercept to allow it to register with Test Hub.


1. Perform the following steps in Test Hub:

a. Log in to Test Hub.

The **Team Space Dashboard** page is displayed.

b. Generate the offline user token by clicking the **User** icon , and then click **Create Token**.

c. Copy the offline user token to a text file.

d. Click the **Settings** icon  of the team space in the **My Team Spaces** page.

The **Team Space Configuration** page is displayed.

e. Copy the text that is displayed under **URL alias** as the *Team Space ID*.

2. Perform the following steps to edit the `registration.xml` file for the intercept. For most intercepts this is in the root directory of the extracted intercept compressed file. For the HTTP/TCT proxy it is in the `<DTVCP_install_directory>\httptcp` directory on the computer on which you have installed Test Virtualization Control Panel.

a. Open the `registration.xml` file by using a text editor.

b. Enter the values for the following attributes:



Note: If an entry exists, you must either comment out that entry and add a new one or replace the contents with the values indicated.

Attribute	Value
server base-url	URL of Test Hub
security-token	Offline user token generated from Test Hub
team-space-id	ID of the team space to which the proxies must register

- c. Save and close the file.
 - d. Perform any of the following actions:
 - Restart the HTTP proxy, if you updated the `registration.xml` file of the HTTP/TCP proxy.
 - Restart the application that is configured to use the any of the other intercepts other than HTTP/TCP.
3. Perform the following steps in Test Hub to view the HTTP proxies that are registered in the team space:
- a. Log in to Test Hub.
 - b. Click **Infrastructure > Agents and Intercepts** to view the registered intercepts.

The intercepts that you configured with your offline user token and *Team Space ID* are displayed.

Results

You have successfully configured intercepts and viewed that they are registered with the team space in Test Hub.

What to do next

You can perform the following tasks:

- Read for the prerequisite tasks that you must perform before you can run the HTTP virtual services. See [Prerequisites for running HTTP virtual services in a Kubernetes cluster on page 742](#).
- Run the virtual service resource. See [Configuring a run of a virtual service on page 750](#).
- View the routing rules for the intercept. See [Viewing routing rules of the virtual services on page 783](#).
- View the routing rules for the intercept in a running instance of the virtual service. See [Viewing configurations of running instances of virtual services on page 775](#).

Viewing intercepts that are registered with a team space on DevOps Test Hub

After you install the HTTP proxies and configure them with your offline user token, the proxies register with HCL DevOps Test Hub (Test Hub) in a team space as intercepts and you can view the registered intercepts on the **Agents and Intercepts** page of the team space.

Before you begin

You must have completed the following tasks:

- Completed the prerequisite tasks before you configure a run of the HTTP virtual services. See [Prerequisites for running HTTP virtual services in a Kubernetes cluster on page 742](#).
- Installed the HTTP proxies of the same version as the version of Test Hub.
- Configured the HTTP proxy with your offline user token so that the proxy can register as an intercept with Test Hub. See [Setting up intercepts on page 744](#).

- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Agents and Intercepts** in the navigation pane.

The **Agents and Intercepts** page is displayed.

You can view the API agents, Dockers, intercepts, Kubernetes clusters, or performance agents that are registered with Test Hub.

3. You can view the intercepts that you own in any of the following ways:

- Search for the intercept by entering the name of the intercept in the **Search** field.



Note: You can enter either the full name or any text that is in the name. The search is enabled for case-sensitive text that you can enter.

- Sort the **Type** column to show the items sorted with the intercepts in the rows at the top of the table.
- Sort the **Agents** column and identify the intercepts by their names or by their owner.

You can view the following details about the intercepts that are registered with Test Hub:

- `Any project` is displayed in the **Projects** column indicating that the intercept can be used by virtual services in any project.
- The status of the intercept is displayed in the **Status** column.

Results

You have viewed the registered intercepts from the **Agents and Intercepts** page.

What to do next

You can start the virtual services that use the registered intercepts. See [Configuring a run of a virtual service on page 750](#).

Viewing virtual service resources

When project repositories contain virtual service resources and you want to view the resources so that you can start them, you can view the virtual services from the **Resources** page in your team space on HCL DevOps Test Hub (Test Hub).


Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created virtual services that use any of the supported transports for tests in Test Integrations and APIs, committed the test resources to the remote repository, and added the repository to your project.

About this task

After you have committed the test assets and resources to the remote repositories and added the repositories to projects in your team space on Test Hub, you can go to the **Resources** page and perform the following tasks:

- View the virtual service resources that are contained in any branch of the repositories that are added to your project.
- View the components in the System Model that are associated with the virtual service resources.
- Create or use saved filters for viewing virtual service resources.
- Run or start an instance of the virtual service resource.
- View the number of instances of the virtual service resources that are running.
- View all instances of a running virtual service on the **Instances** page by clicking the **Show in instances page** icon .

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the virtual service resources in the test assets by clicking **Projects > My Projects > project_name**.




The **Overview** page is displayed.


3. Click **Virtualization > Resources** in the navigation pane.

The **Resources** page is displayed.

4. Complete the steps for the task that you want to perform as listed in the following table:

Task	Action
Viewing the virtual service resources that are contained in a branch of any of the repositories that are added to your project.	Select the branch of the repository that contains the virtual services that you want to run from the list in the Branch field. All virtual services in the selected branch are displayed on the Resources page.
Viewing the components that are associated with the virtual service resource.	The components that are associated with the virtual service resource are displayed in the Components column.

Task	Action
	Click the component in the Components column to view the virtual resource on the System Model page.
Creating or using saved filters for viewing specific instances of virtual service resources.	<p>To create a query, perform the following actions:</p> <ol style="list-style-type: none"> Click New filter. Create a rule with an appropriate operator. Select criteria such as <i>Name, Path, Instance number, Instance state, or Components</i>. Select the condition, and then enter the value for the criteria. Apply the filter query. <p>The virtual services that match the filter criteria are displayed.</p> <ol style="list-style-type: none"> Save the filter query for retrieving it from the saved filters list. <p>To use a saved filter query, perform the following actions:</p> <p> Note: To open the filter query, you must have created and saved a filter query.</p> <ol style="list-style-type: none"> Click the Open filters icon  Select the saved filter. Apply the filter. <p>The virtual services that match the filter criteria are displayed.</p>
Running an instance of the virtual service resource.	<p>After you identify the virtual service that you want to run, perform the following actions:</p> <ol style="list-style-type: none"> Click the Execute icon  in the row of the identified virtual service. <p>The Execute virtual service dialog is displayed.</p> <ol style="list-style-type: none"> Run the virtual service with the settings that are configured in the resource by clicking Execute or modify any of the settings, and then click Execute. <p>An instance of the virtual service starts to run. The virtual service is displayed with the state as <i>Running</i> along with the number of instances that are running in the Active instances column.</p>

Task	Action
Viewing the number of instances of the virtual service resources that are running.	<p>If any of the virtual services in the project repositories are started by any project member with the <i>Tester</i> or <i>Owner</i> role, you can view the number of instances that are running or active from the Resources page.</p> <p>The virtual services are displayed with the state as <code>Running</code> along with the number of instances that are running, in the Active instances column.</p>
Viewing all instances of a running virtual service.	<p>Click the Show in instances page icon  in the row of the virtual service on the Resources page.</p> <p>The Instances page is displayed with all the running instances of a particular virtual service.</p> <p>You can also view the number of requests that are received by the virtual service. The number of requests is displayed as the number of hits when you hover the cursor over the text in the Activity column.</p>

Results

You have viewed virtual services from the **Resources** page on Test Hub.

Related information

[Configuring a run of a virtual service on page 750](#)

Configuring a run of a virtual service

When you have virtual service resources that are in the repositories added to your project on HCL DevOps Test Hub (Test Hub), you must configure a run of the virtual service to start an instance of the virtual service in the Test Hub cluster, a remote Kubernetes cluster, or an API agent.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created virtual services that use any of the supported transports for tests in Test Integrations and APIs, committed the test resources to the remote repository, and added the repository to your project.

- Read about the virtual services that are supported on Test Hub. See [Virtual services that are supported on DevOps Test Hub on page 91](#).
- Completed the prerequisite tasks before you configure a run of the HTTP virtual services. See [Prerequisites for running HTTP virtual services in a Kubernetes cluster on page 742](#).
- Read and completed the tasks mentioned in [Considerations for using a remote Kubernetes cluster on page 417](#), if you want to use a remote Kubernetes cluster as a location to run the test.
- Read and completed the tasks mentioned in [Considerations for running virtual services on remote API agents on page 395](#), if you want to use a remote API agent as the location to run the virtual services.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the virtual service resources in the test assets by clicking **Projects > My Projects > *project_name***.

The **Overview** page is displayed.

3. Click **Virtualization > Resources** in the navigation pane.

The **Resources** page is displayed.


4. Select the branch of the repository that contains the virtual services that you want to run from the list in the **Branch** field.

All virtual services in the selected branch are displayed on the **Resources** page.

5. Identify the virtual service by completing any of the following steps:

- Search for the virtual service by entering the name or the path of the virtual service in the repository in the **Search** field box.
- Create a filter query by using the **New filter** option and complete the following steps:
 - a. Create a rule with an appropriate operator.
 - b. Select criteria such as *Name*, *Path*, *Instance number*, or *Instance state*. Select the condition and enter the value for the criteria. Apply the filter query.

The virtual services that match the filter criteria are displayed.

- Retrieve a saved filter by using the **Open filters** icon  by completing the following steps:





Note: To open the filter query, you must have created and saved a filter query.

- a. Select the saved filter.
- b. Apply the filter.

The virtual services that match the filter criteria in the filter that is applied are displayed.

You have identified the virtual service resources that you want to run.

6. Click the **Open action menu** icon  in the row of the identified virtual service, and then click **Execute** icon .

The **Execute virtual service** dialog is displayed.

7. Select the version of the virtual service that is in the repository that you want to start by performing any of the following actions:

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version committed followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

8. Select the environment that was used to bind the physical and logical resource in the API project, in the **ENVIRONMENT** tab.



Important: The configuration that you set for the run in the **Execute virtual service** dialog is preserved when you run the same virtual service again. The configurations that you set are not available to other members when they want to run the virtual service. For example, if you selected an environment, the same environment is selected when you run the virtual service again.

9. Enter a label, if required.

A label that you enter for the test run that helps you to identify the virtual service instance on the **Instances** page. The label that you entered is displayed for the virtual service under the **Labels** column on the **Instances** page. After you have created a label, any member of the project can use that label.

10. Follow the instructions if you want to modify the configurations for the behavior of the virtual service:

- a. Click the **BEHAVIOR** tab, if it is not already open.
- b. Configure or change the settings for the following options:



Note: The settings displayed are the settings that were configured for the virtual service when it was authored in Test Integrations and APIs.



Option	Description						
Performance	<p>The following settings are available for handling of requests by the virtual service:</p> <table border="1" data-bbox="574 533 1417 1654"> <thead> <tr> <th data-bbox="574 533 891 590">Option</th> <th data-bbox="891 533 1417 590">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="574 590 891 1482">Optimize performance</td> <td data-bbox="891 590 1417 1482"> <p>If enabled, attempts are made to reduce the amount of processing between the time the virtual service receives a request and the time it sends a response. Specific optimizations depend on the message contents, as in the following examples:</p> <ul style="list-style-type: none"> ▪ When the virtual service receives requests, all validations are disabled, and for all XML payloads, the store and filter actions are converted to use XPath expressions. ▪ When the virtualization sends responses, any store actions set on a message are disabled, and any XML content is collapsed when the virtual service is compiled instead of being collapsed every time that a response is sent. <p>The optimization is disabled as the default action for this setting and you can enable the performance optimization if you fully understand the implications of optimizing the performance.</p> </td> </tr> <tr> <td data-bbox="574 1482 891 1654">Threads</td> <td data-bbox="891 1482 1417 1654"> <p>The maximum number of threads used in processing requests received by the virtual service. The default number of threads is <i>10</i>.</p> </td> </tr> </tbody> </table>	Option	Description	Optimize performance	<p>If enabled, attempts are made to reduce the amount of processing between the time the virtual service receives a request and the time it sends a response. Specific optimizations depend on the message contents, as in the following examples:</p> <ul style="list-style-type: none"> ▪ When the virtual service receives requests, all validations are disabled, and for all XML payloads, the store and filter actions are converted to use XPath expressions. ▪ When the virtualization sends responses, any store actions set on a message are disabled, and any XML content is collapsed when the virtual service is compiled instead of being collapsed every time that a response is sent. <p>The optimization is disabled as the default action for this setting and you can enable the performance optimization if you fully understand the implications of optimizing the performance.</p>	Threads	<p>The maximum number of threads used in processing requests received by the virtual service. The default number of threads is <i>10</i>.</p>
Option	Description						
Optimize performance	<p>If enabled, attempts are made to reduce the amount of processing between the time the virtual service receives a request and the time it sends a response. Specific optimizations depend on the message contents, as in the following examples:</p> <ul style="list-style-type: none"> ▪ When the virtual service receives requests, all validations are disabled, and for all XML payloads, the store and filter actions are converted to use XPath expressions. ▪ When the virtualization sends responses, any store actions set on a message are disabled, and any XML content is collapsed when the virtual service is compiled instead of being collapsed every time that a response is sent. <p>The optimization is disabled as the default action for this setting and you can enable the performance optimization if you fully understand the implications of optimizing the performance.</p>						
Threads	<p>The maximum number of threads used in processing requests received by the virtual service. The default number of threads is <i>10</i>.</p>						
Operation	Specifies an operation referenced by the virtual service.						
Response time	Specifies the response time behavior for responses sent by the virtual service for the selected operation.						

Option	Description										
	<p>You can modify the value by selecting a value from the following options:</p> <table border="1" data-bbox="574 321 1419 949"> <thead> <tr> <th data-bbox="574 321 727 378">Option</th> <th data-bbox="727 321 1419 378">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="574 378 727 472">No delay</td> <td data-bbox="727 378 1419 472">Select this option for a response with no delay.</td> </tr> <tr> <td data-bbox="574 472 727 604">Minimum delay</td> <td data-bbox="727 472 1419 604">Select this option for a delay in the response by entering the required delay in milliseconds (ms).</td> </tr> <tr> <td data-bbox="574 604 727 779">Uniform distribution</td> <td data-bbox="727 604 1419 779">Select this option for a uniformly distributed response time by specifying the minimum and maximum delay in milliseconds (ms).</td> </tr> <tr> <td data-bbox="574 779 727 949">Gaussian distribution</td> <td data-bbox="727 779 1419 949">Select this option for a response time with Gaussian distribution by specifying the minimum and maximum delay in milliseconds (ms).</td> </tr> </tbody> </table>	Option	Description	No delay	Select this option for a response with no delay.	Minimum delay	Select this option for a delay in the response by entering the required delay in milliseconds (ms).	Uniform distribution	Select this option for a uniformly distributed response time by specifying the minimum and maximum delay in milliseconds (ms).	Gaussian distribution	Select this option for a response time with Gaussian distribution by specifying the minimum and maximum delay in milliseconds (ms).
Option	Description										
No delay	Select this option for a response with no delay.										
Minimum delay	Select this option for a delay in the response by entering the required delay in milliseconds (ms).										
Uniform distribution	Select this option for a uniformly distributed response time by specifying the minimum and maximum delay in milliseconds (ms).										
Gaussian distribution	Select this option for a response time with Gaussian distribution by specifying the minimum and maximum delay in milliseconds (ms).										
Passthrough	<p>Specifies the pass through behavior for the selected operation when requests are not handled within the virtual service.</p> <p>You can modify the value by selecting a value from the following options:</p> <table border="1" data-bbox="574 1161 1419 1493"> <tbody> <tr> <td data-bbox="574 1161 727 1304">Discard</td> <td data-bbox="727 1161 1419 1304">This option stops the system under test from receiving the intercepted message. This option can disrupt the calling system. For example, the system might time out while it waits for a reply.</td> </tr> <tr> <td data-bbox="574 1304 727 1398">Pass Through</td> <td data-bbox="727 1304 1419 1398">This option passes the intercepted message to the system under test, with an optional delay.</td> </tr> <tr> <td data-bbox="574 1398 727 1493">Simulate Error</td> <td data-bbox="727 1398 1419 1493">This option returns an error to the calling system. The message is not passed to the system under test.</td> </tr> </tbody> </table>	Discard	This option stops the system under test from receiving the intercepted message. This option can disrupt the calling system. For example, the system might time out while it waits for a reply.	Pass Through	This option passes the intercepted message to the system under test, with an optional delay.	Simulate Error	This option returns an error to the calling system. The message is not passed to the system under test.				
Discard	This option stops the system under test from receiving the intercepted message. This option can disrupt the calling system. For example, the system might time out while it waits for a reply.										
Pass Through	This option passes the intercepted message to the system under test, with an optional delay.										
Simulate Error	This option returns an error to the calling system. The message is not passed to the system under test.										

11. Follow the instructions, if the virtual service references datasets.

- You can use the dataset referenced in the asset.
- You can choose to override the dataset with another data source. If alternative data sources are available, select from the set of overrides available.

12. Follow the instructions if the virtual service requires a variable that must be passed at run time.

- a. Click the **VARIABLES** tab, if it is not already open.
- b. Choose one of the following methods to add the variables:
 - To add new variables manually, click the **Add Variable** icon , enter the name, and value of the variable.
 - To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon  and select the **Upload from local system** or **Browse from server** to select the variable file.



Note: You must have created a file with the variables before you can select the file.

13. Click **Advanced** to make the following advanced configurations:

- a. Enter any JVM arguments that must be passed at run time in the **JVM Arguments** field.



Note: Each JVM argument should be separated with white space.

For example, you can set a maximum Java heap size.

- b. Enter the environment variables that must be passed at run time in the **Environment Variables** field, if applicable.

For example, enter the environment variables when the third-party libraries that are used in the run refer to the environment variables for configuration.

- c. Select the stub logging level for the virtual service from the following options in the **Logging** list:

Option	Description
None	Specifies that the virtual service does not write log messages.
Normal	Specifies that the virtual service writes informational messages.
Debug	Specifies that the virtual service writes informational and debugging messages.

- d. Enter other configuration options as parameters and their values in the **Additional Configuration Parameters** fields, if applicable.

You can refer to the additional parameters that you can use for virtual services from the topic in the related links.

For example, if you want to start the virtual service to run in a new container, you can specify the following parameter and its value:

Parameter name	Value
<code>stub.dedicated.container</code>	<code>true</code>



Note: Click **Add** to add additional parameters.



Note: You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for the fields for the advanced settings is null or an empty field.



Notes:

- If you have configured some or all of the settings for the current run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you want to repeat a run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

14. Select the step for the location to run the virtual services based on the following types of virtual services:

- If you want to run any of the following types of virtual services only in the default Kubernetes cluster or a remote Kubernetes cluster, then go to [Step 15 on page 756](#).
- If you want to run the virtual services on a remote API agent that you added to your project, then go to [Step 16 on page 757](#).

15. Follow the instructions if you want to change the location of a Kubernetes cluster for running the test:

- a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the virtual service resource runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.



Important: You must have added the registered remote Kubernetes clusters to your project that are then displayed under the **Override** column:



Notes:



- If the remote clusters are not added to your project, the option **No override options** is displayed as the default value and the virtual service resource runs in the Kubernetes cluster of Test Hub.
- If remote clusters are added to your project, the added clusters are displayed along with their availability status and ownership information.

b. Select the location where you want to run the virtual service resource from the following options:

- Select the **Default Cluster** when no remote clusters are available in your project.
- Select the remote cluster from the list when a remote cluster is available in your project.
- Select **No override options**, if you selected any remote cluster and want to revert to the **Default Cluster** to run the virtual service resource.

c. Click **Execute**.

The run is initiated.

16. Follow the instructions if you want to run the virtual services on the remote API agent that is connected to the default Kubernetes cluster:

a. Click the **LOCATION** tab, if it is not already open.

The API agents from the list of registered API agents that match with the attributes or tags configured for the virtual service resources in the project are listed under the **Host** column. The information about the availability of the agent is displayed.



Note: You must have added the API agents to your project from the **Agents and Intercepts** page for the API agents to be displayed under the **Override** column.

The default value for the API agents is null or an empty field if no agents were configured in the project resources. If the project resources contains API agents that match with the agent tags, then the default API agent is the first item to be displayed on the list of API agents listed in the increasing alphabetical order.

b. Select the API agent where you want to run the virtual service in the following scenarios:

If...	Then...	Action
The API agent that is specified in the project resources is available and no other agents are added to the project.	The API agent that is specified in the project resources is displayed in the Override column.	Select the API agent in the Host that is also displayed in the Override column as the location to run the test.
The API agent that is specified in the project resources is	There is no API agent displayed for override in the Override column.	You cannot run the test on the API agent.

If...	Then...	Action
not available and no other API agents are added to the project.		You must wait until the API agent is available to run the test on the API agent.
The API agent that is specified in the project resources is available and other API agents added to the project.	<p>API agents that have capabilities that match with the API agent capabilities specified in the project resources are listed in the Override column as follows:</p> <ul style="list-style-type: none"> ▪ API agents with capabilities that best match the capabilities of the API agent in the project resources are at the top of the list. ▪ API agents with capabilities that do not match with the capabilities of the API agent in the project resources are listed subsequently. <p>You can find details of both API agents for the matching capabilities when you hover over the agent in the Override column.</p>	<p>Perform any of the following actions:</p> <ul style="list-style-type: none"> ▪ Select the API agent in the Host that is also displayed in the Override column as the location to run the test. ▪ Select an API agent in the Override column as the location to run the test.

c. Click **Execute**.


The test run is initiated.

Results

You have started a virtual service from the **Resources** page on Test Hub.

What to do next

After you start a run of a virtual service, you can view the status of the virtual service in any of the following ways:

- On the **Resources** page, the virtual services are displayed with the state as `Running` along with the number of instances that are running in the **Active instances** column.
- On the **Resources** page, you must click the **Show in instances page** icon  in the row of the virtual service to view all the running instances of a particular virtual service on the **Instances** page.
- Go to the **Instances** page by clicking **Virtualization > Instances**. You can view instances of all the virtual services that are running and are displayed with the state as `Running` under the **State** column. You can also view the requests that are received by the virtual service. The number of requests is displayed as number of hits when you hover the cursor over the text in the **Activity** column.

You can view the details, configuration settings, routing rules, usage statistics, or logs of a specific running instance of the virtual service. See [Viewing configurations of running instances of virtual services on page 775](#).

You can modify the behavior or the logging level of a running instance of the virtual service that you started before running tests on them. See [Modifying configurations of running instances of virtual services on page 780](#).

When you have completed testing the virtual service, you must stop the running virtual service. See [Stopping virtual services on page 787](#).

Related reference

[Additional configuration parameters for virtual services on page 759](#)


Additional configuration parameters for virtual services





You can find information about the additional configuration parameters that you can set when you want to run virtual services on Test Hub.


When you use the server UI to start the virtual service, some of the parameters are handled by fields in the **Execute virtual service** dialog.


You can enter the other parameters as a *name-value* pair in the **Additional configuration parameters** field that is displayed in the **Advanced** settings panel of the **Execute virtual service** dialog.



The following additional configuration parameters are supported when you configure a run for a virtual service.

Requirement	Configuration parameter name	Supported values	An example value	Result of using the example value
To specify a unique identifier for use in the generated host name.	<code>stub.ingress.host.identifier</code>	Any <code><custom_id></code> that you specify.  Note: The <code><custom_id></code> must start and end with	<code>mystub</code>	See Running HTTP virtual services in Kubernetes cluster without using proxies on page 765 , to know

Requirement	Configuration parameter name	Supported values	An example value	Result of using the example value
<p> Note: This parameter applies only if the virtual service is run in a Kubernetes cluster and not on an API agent.</p>		<p> an alphanumeric character. The characters supported include a-z, 0-9, and a hyphen. No other characters can be used in the <code><custom_id></code>. The <code><custom_id></code> must not be longer than 32 characters.</p>		<p>how the identifier you specified is used.</p>
<p>To specify whether a virtual service instance should be run in a dedicated container.</p> <p> Note: This parameter applies only if the virtual service is run in a Kubernetes cluster and not on an API agent.</p>	<p><code>stub.dedicated.container</code></p>	<ul style="list-style-type: none"> • <i>true</i> • <i>false</i> <p> Note: The default value is <i>false</i>.</p>	<p><i>true</i></p>	<p>A dedicated container is used to run a single virtual service instance.</p>
<p>To specify whether performance optimization is enabled</p>	<p><code>stub.performance.optimize</code></p>	<ul style="list-style-type: none"> • <i>true</i> • <i>false</i> 	<p><i>true</i></p>	<p>The performance optimization is enabled for the virtual service.</p>

Requirement	Configuration parameter name	Supported values	An example value	Result of using the example value
<p>for the virtual service instance. The value set overrides the setting from the stub definition.</p> <p>This option can be set in the Behavior tab from the UI.</p>		 Note: The default value is <i>false</i> .		
<p>To specify the number of threads to be used for processing requests for the virtual service instance. The value set overrides the setting from the stub definition.</p> <p>This option can be set in the Behavior tab from the UI.</p>	<code>stub.worker.thread-count</code>	Any number that you specify as number of threads to use.	15	15 threads are used for processing requests for the virtual service instance.
<p>To specify the stub logging level for the virtual service instance. The value set overrides the setting from the stub definition.</p> <p>This option can be set in the Advanced settings from the UI.</p>	<code>stub.logging.level</code>	<ul style="list-style-type: none"> • <i>none</i> • <i>normal</i> • <i>debug</i> 	<i>debug</i>	The virtual service writes informational and debugging messages to the container log.
<p>To specify the pass-through behaviour type of the virtual service instance. The value set overrides</p>	<code>stub.pass.through-behavior</code>	Valid values depend on the stub operation transport type but can be as follows:	<i>simulate_error</i>	This option returns an error to the calling system. The message is not passed to the system under test.

Requirement	Configuration parameter name	Supported values	An example value	Result of using the example value
<p>the setting from the stub definition.</p> <p>This option can be set in the Behavior tab from the UI.</p>		<ul style="list-style-type: none"> • <i>simulate_error</i> • <i>discard</i> • <i>pass_through</i> 		
<p>To specify the delay in passing through from the virtual service instance. The value set overrides the setting from the stub definition.</p>	<code>stub.pass.through.delay.period</code>	Any number of seconds in milliseconds.	<i>10000</i>	A delay of 10 seconds in passing through from the virtual service instance.
<p>To specify the status code to use in the virtual service instance response when simulating an error. The value set overrides the setting from the stub definition.</p> <p> Note: Applies to HTTP stubs.</p> <p>This option can be set in the Behavior tab from the UI.</p>	<code>stub.pass.through.status.code</code>	You must specify a valid HTTP status code as the value. For example, <i>503</i> .	<i>500</i>	The HTTP error code of <i>500</i> is sent in the response if the virtual service simulates an error when passing through.
<p>To specify the reason phrase to use in the virtual service instance response when simulating an error. The value set overrides the setting from the stub definition.</p>	<code>stub.pass.through.reason.phrase</code>	You must specify the text of the status message as the value.	<i>my custom error msg</i>	The text that you specify is set as the HTTP reason phrase in the response if the virtual service simulates an error when passing through.

Requirement	Configuration parameter name	Supported values	An example value	Result of using the example value
<p> Note: Applies to HTTP stubs.</p> <p>This option can be set in the Behavior tab from the UI.</p>				
<p>To specify the type of response-time behavior that is required for the virtual service instance. The value set overrides the setting from the stub definition.</p> <p>This option can be set in the Behavior tab from the UI.</p>	<pre>stub.response.time- .type</pre>	<ul style="list-style-type: none"> • <i>no_delay</i> • <i>minimum_delay</i> • <i>gaussian</i> • <i>uniform</i> • <i>performance_profile</i> <p> Note: <i>performance_profile</i> is only valid if the stub operation in the stub definition is defined to use a performance profile.</p>	<pre>no_delay</pre>	<p>The virtual service sends a response without any delay.</p>
<p>To specify the minimum time for the response time behaviour required for the virtual service instance. The value set overrides the setting from the stub definition.</p>	<pre>stub.response.time- .minimum</pre>	<p>Any number of seconds in milliseconds.</p>	<pre>5000</pre>	<p>The virtual service sends a response after a minimum delay of 5 seconds.</p>

Requirement	Configuration parameter name	Supported values	An example value	Result of using the example value
This option can be set in the Behavior tab from the UI.				
To specify the maximum time for the response time behaviour required for the virtual service instance. The value set overrides the setting from the stub definition. This option can be set in the Behavior tab from the UI.	<code>stub.response.time-maximum</code>	Any number of seconds in milliseconds.	<code>20000</code>	The virtual service must send a response after a maximum delay of 20 seconds.
To override the value of for an input tag used by the virtual service.	<code>stub.input.tag.name</code> , where <i>name</i> is the name of the input tag that you want the virtual service to use.	The value to set for the input tag.	You can specify a tag name as <code>stub.input-tag.mytag</code> with a value of <code>red</code> .	The input tag named <i>mytag</i> will have the value <i>red</i> when the virtual service is run.

Virtual services where the stub definition refers to multiple operations

When a stub definition has two or more events that refer to different operations and each operation has a different behavior, you must specify the `<operation_id>` suffixed to the configuration parameter name in the **Additional configuration parameters** field.



Note: The `<operation_id>` is the internal ID of the operation that is found in the **Documentation** tab when you open the operation in Test Integrations and APIs.

When the stub definition refers to multiple operations and you want to start a run of a virtual service, you must append the `<operation_id>` to the `<parameter_name>` in the **Additional configuration parameters** field.

The configuration parameter names that you must use for the virtual service, which is associated with a particular operation are as follows:

- `stub.pass.through.behavior.<operation_id>`
- `stub.pass.through.delay.period.<operation_id>`

- `stub.pass.through.status.code.<operation_id>`
- `stub.pass.through.reason.phrase.<operation_id>`
- `stub.response.time.type.<operation_id>`
- `stub.response.time.<operation_id>`
- `stub.response.time.maximum.<operation_id>`

For example, if the `<operation_id>` is `7281b750:162483a09f1:-7e68`, the parameter name for the `stub.response.time.type` option that you must specify is `stub.response.time.type.7281b750:162483a09f1:-7e68`.

Related information

[Configuring a run of a virtual service on page 750](#)

Running HTTP virtual services in Kubernetes cluster without using proxies

When you run HTTP virtual services in the Kubernetes cluster of HCL DevOps Test Hub (Test Hub) or a remote Kubernetes cluster, the endpoint on which the virtual service is exposed externally uses a host name that includes a generated ID. If a client application is required to call the virtual service directly via its external hostname and not via the HTTP proxy, then you might want to determine the hostname to configure the client first before you start the virtual service.



Important: This task does not apply when you use an API agent to run the HTTP virtual services.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in a team space on Test Hub and added the repository that contains the HTTP virtual services.
- Completed the prerequisite tasks before you start HTTP virtual services. See [Prerequisites for running HTTP virtual services in a Kubernetes cluster on page 742](#).
- Selected the branch of the Git repository to view test resources on the **Execution** page.

About this task

When you installed Test Hub on Ubuntu, the default HTTP virtual service endpoints are of the following form:

```
in-<unique_id>.<ingress_domain>
```

For example, with an ingress domain as **10.1.2.3.nip.io**, a virtual service endpoint might be as follows:

```
in-cd3a755635574c6fa7e264911301821a.10.1.2.3.nip.io
```

When you installed Test Hub 10.2.1 or earlier on Red Hat OpenShift, the default HTTP virtual service endpoints are of the following form:

```
<ingress_domain_first_label>-<unique_id>.<remainder_of_ingress_domain>
```

For example, with an ingress domain as **dth.mycluster-123456-0000.region.containers.appdomain.cloud**, a virtual service endpoint might be as follows:

```
dth-cd3a755635574c6fa7e264911301821a.mycluster-123456-0000.region.containers.appdomain.cloud
```

The **<unique_id>** is generated when the virtual service is started. The full hostname can be viewed in the **Routing to** field of the routing rule created for the virtual service on the **Intercept Rules** page. When traffic is routed to the virtual service via the HTTP proxy, a client application is unaware of this hostname. If the client application is required to call the virtual services directly, it must be aware of the hostname of the virtual service.

To allow a hostname to be known before you start the virtual service, you can replace the generated ID with a user defined value that is specified as an additional configuration parameter when starting the stub. This value is referred to as **<custom_id>**.



Notes:

- The value of the **<custom_id>** that you specify is used as the **<unique_id>** for the virtual service.
- The **<custom_id>** must start and end with an alphanumeric character. The characters supported include a-z, 0-9, and a hyphen. No other characters can be used in the **<custom_id>**. The **<custom_id>** must not be longer than 32 characters.



Restriction: The **<custom_id>** must be unique to a virtual service that is running. You must not assign the same **<custom_id>** to another virtual service that is running simultaneously. Doing so might result in an undefined behavior of the virtual service.

For example, on Ubuntu, if you specify the value of the **<custom_id>** for the virtual service as `stub123-test` and Test Hub has an ingress domain as **10.1.2.3.nip.io**. Then the hostname of the virtual service endpoint is as follows:

```
in-stub123-test.10.1.2.3.nip.io
```

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the virtual service resources in the test assets by clicking **Projects > My Projects > *project_name***.



The **Overview** page is displayed.

3. Click **Virtualization > Resources** in the navigation pane.

The **Resources** page is displayed.

4. Select the branch of the repository that contains the virtual services that you want to run from the list in the **Branch** field.

All virtual services in the selected branch are displayed on the **Resources** page.

5. Identify the virtual service that you want to run.
6. Click the **Open action menu** icon  in the row of the identified virtual service, and then click **Execute** icon .

The **Execute virtual service** dialog is displayed.

7. Select the version of the virtual service that is in the repository that you want to start by performing any of the following actions:

- Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:


- Commit message.
- Tags labeled for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version committed followed by the versions committed previously.

- Expand the list in the **Version** field, and then search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and it is selected for the test run.

8. Select the environment that was used to bind the physical and logical resource in the API project, in the **ENVIRONMENT** tab.

 **Important:** The configuration that you set for the run in the **Execute virtual service** dialog is preserved when you run the same virtual service again. The configurations that you set are not available to other members when they want to run the virtual service. For example, if you selected an environment, the same environment is selected when you run the virtual service again.

9. Enter a label, if required.

A label that you enter for the test run that helps you to identify the virtual service instance on the **Instances** page. The label that you entered is displayed for the virtual service under the **Labels** column on the **Instances** page. After you have created a label, any member of the project can use that label.

10. Follow the instructions if you want to modify the configurations for the behavior of the virtual service:

- a. Click the **BEHAVIOR** tab, if it is not already open.
- b. Configure or change the settings for the following options:



Note: The settings displayed are the settings that were configured for the virtual service when it was authored in Test Integrations and APIs.

Option	Description						
Performance	<p>The following settings are available for handling of requests by the virtual service:</p> <table border="1" data-bbox="574 661 1419 1785"> <thead> <tr> <th data-bbox="574 661 891 716">Option</th> <th data-bbox="891 661 1419 716">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="574 716 891 1610">Optimize performance</td> <td data-bbox="891 716 1419 1610"> <p>If enabled, attempts are made to reduce the amount of processing between the time the virtual service receives a request and the time it sends a response. Specific optimizations depend on the message contents, as in the following examples:</p> <ul style="list-style-type: none"> ▪ When the virtual service receives requests, all validations are disabled, and for all XML payloads, the store and filter actions are converted to use XPath expressions. ▪ When the virtualization sends responses, any store actions set on a message are disabled, and any XML content is collapsed when the virtual service is compiled instead of being collapsed every time that a response is sent. <p>The optimization is disabled as the default action for this setting and you can enable the performance optimization if you fully understand the implications of optimizing the performance.</p> </td> </tr> <tr> <td data-bbox="574 1610 891 1785">Threads</td> <td data-bbox="891 1610 1419 1785"> <p>The maximum number of threads used in processing requests received by the virtual service. The default number of threads is <i>10</i>.</p> </td> </tr> </tbody> </table>	Option	Description	Optimize performance	<p>If enabled, attempts are made to reduce the amount of processing between the time the virtual service receives a request and the time it sends a response. Specific optimizations depend on the message contents, as in the following examples:</p> <ul style="list-style-type: none"> ▪ When the virtual service receives requests, all validations are disabled, and for all XML payloads, the store and filter actions are converted to use XPath expressions. ▪ When the virtualization sends responses, any store actions set on a message are disabled, and any XML content is collapsed when the virtual service is compiled instead of being collapsed every time that a response is sent. <p>The optimization is disabled as the default action for this setting and you can enable the performance optimization if you fully understand the implications of optimizing the performance.</p>	Threads	<p>The maximum number of threads used in processing requests received by the virtual service. The default number of threads is <i>10</i>.</p>
Option	Description						
Optimize performance	<p>If enabled, attempts are made to reduce the amount of processing between the time the virtual service receives a request and the time it sends a response. Specific optimizations depend on the message contents, as in the following examples:</p> <ul style="list-style-type: none"> ▪ When the virtual service receives requests, all validations are disabled, and for all XML payloads, the store and filter actions are converted to use XPath expressions. ▪ When the virtualization sends responses, any store actions set on a message are disabled, and any XML content is collapsed when the virtual service is compiled instead of being collapsed every time that a response is sent. <p>The optimization is disabled as the default action for this setting and you can enable the performance optimization if you fully understand the implications of optimizing the performance.</p>						
Threads	<p>The maximum number of threads used in processing requests received by the virtual service. The default number of threads is <i>10</i>.</p>						
Operation	Specifies an operation referenced by the virtual service.						

Option	Description										
Response time	<p>Specifies the response time behavior for responses sent by the virtual service for the selected operation.</p> <p>You can modify the value by selecting a value from the following options:</p> <table border="1" data-bbox="574 457 1416 1087"> <thead> <tr> <th data-bbox="574 457 727 510">Option</th> <th data-bbox="727 457 1416 510">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="574 510 727 604">No delay</td> <td data-bbox="727 510 1416 604">Select this option for a response with no delay.</td> </tr> <tr> <td data-bbox="574 604 727 741">Minimum delay</td> <td data-bbox="727 604 1416 741">Select this option for a delay in the response by entering the required delay in milliseconds (ms).</td> </tr> <tr> <td data-bbox="574 741 727 915">Uniform distribution</td> <td data-bbox="727 741 1416 915">Select this option for a uniformly distributed response time by specifying the minimum and maximum delay in milliseconds (ms).</td> </tr> <tr> <td data-bbox="574 915 727 1087">Gaussian distribution</td> <td data-bbox="727 915 1416 1087">Select this option for a response time with Gaussian distribution by specifying the minimum and maximum delay in milliseconds (ms).</td> </tr> </tbody> </table>	Option	Description	No delay	Select this option for a response with no delay.	Minimum delay	Select this option for a delay in the response by entering the required delay in milliseconds (ms).	Uniform distribution	Select this option for a uniformly distributed response time by specifying the minimum and maximum delay in milliseconds (ms).	Gaussian distribution	Select this option for a response time with Gaussian distribution by specifying the minimum and maximum delay in milliseconds (ms).
Option	Description										
No delay	Select this option for a response with no delay.										
Minimum delay	Select this option for a delay in the response by entering the required delay in milliseconds (ms).										
Uniform distribution	Select this option for a uniformly distributed response time by specifying the minimum and maximum delay in milliseconds (ms).										
Gaussian distribution	Select this option for a response time with Gaussian distribution by specifying the minimum and maximum delay in milliseconds (ms).										
Passthrough	<p>Specifies the pass through behavior for the selected operation when requests are not handled within the virtual service.</p> <p>You can modify the value by selecting a value from the following options:</p> <table border="1" data-bbox="574 1297 1416 1627"> <tbody> <tr> <td data-bbox="574 1297 719 1434">Discard</td> <td data-bbox="719 1297 1416 1434">This option stops the system under test from receiving the intercepted message. This option can disrupt the calling system. For example, the system might time out while it waits for a reply.</td> </tr> <tr> <td data-bbox="574 1434 719 1535">Pass Through</td> <td data-bbox="719 1434 1416 1535">This option passes the intercepted message to the system under test, with an optional delay.</td> </tr> <tr> <td data-bbox="574 1535 719 1627">Simulate Error</td> <td data-bbox="719 1535 1416 1627">This option returns an error to the calling system. The message is not passed to the system under test.</td> </tr> </tbody> </table>	Discard	This option stops the system under test from receiving the intercepted message. This option can disrupt the calling system. For example, the system might time out while it waits for a reply.	Pass Through	This option passes the intercepted message to the system under test, with an optional delay.	Simulate Error	This option returns an error to the calling system. The message is not passed to the system under test.				
Discard	This option stops the system under test from receiving the intercepted message. This option can disrupt the calling system. For example, the system might time out while it waits for a reply.										
Pass Through	This option passes the intercepted message to the system under test, with an optional delay.										
Simulate Error	This option returns an error to the calling system. The message is not passed to the system under test.										

11. Click **Advanced** to make the following advanced configurations:

- a. Enter the following configuration options as parameters and their values in the **Additional Configuration Parameters** fields.

Parameter name	Value
stub.ingress.host.identifier	<custom_id>

12. Click **Execute**.


The run is initiated.

Results

You have started an HTTP virtual service that does not use a proxy. The virtual service exposes the hostname that contains a custom ID that you specified.

What to do next

After you start a run of a virtual service, you can view the status of the virtual service in any of the following ways:

- On the **Resources** page, the virtual services are displayed with the state as `Running` along with the number of instances that are running in the **Active instances** column.
- On the **Resources** page, you must click the **Show in instances page** icon  in the row of the virtual service to view all the running instances of a particular virtual service on the **Instances** page.
- Go to the **Instances** page by clicking **Virtualization > Instances**. You can view instances of all the virtual services that are running and are displayed with the state as `Running` under the **State** column. You can also view the requests that are received by the virtual service. The number of requests is displayed as number of hits when you hover the cursor over the text in the **Activity** column.

Viewing running instances of virtual services

After you start virtual services from the **Resources** page, you can view the instances of the virtual services that are running from the **Instances** page on HCL DevOps Test Hub (Test Hub). You can view the details, configuration settings, routing rules, usage statistics, or logs of a specific running instance of the virtual service. You can also stop the running instance.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Started the virtual service resources as a *Tester* or *Owner* from the **Resources** page.

About this task

After you started the virtual services from the **Resources** page on Test Hub, you can go to the **Instances** page and you can perform the following tasks:

- View all running instances of the virtual services that are contained in the project repositories.
- View the components in the system model that are associated with the virtual service instances.
- Search for a specific instance of the virtual service for which you want to view the details.
- Create or use saved filters for viewing specific running instances of the virtual service.
- View the details, configuration settings, routing rules, usage statistics, or logs of a specific running instance of the virtual service.
- Change the behavior or logging level of a running instance of the virtual service before you run tests on the virtual service.
- Stop a specific running instance of the virtual service.
- Stop multiple instances or all instances of the virtual service that are running.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the virtual service resources in the test assets by clicking **Projects > My Projects > *project_name***.

The **Overview** page is displayed.

3. Click **Virtualization > Instances** in the navigation pane.

The **Instances** page is displayed.

Any one of the following views is displayed:

- No instances of virtual services are displayed if there are no instances that are running or were stopped in the past hour.
- All running instances of virtual services in your project are displayed.
- No running instances of virtual services are displayed but the option **Show all** is enabled. You can click **Show all** to display the instances that were stopped in the past hour. Instances that were hidden from the display when **Hide inactive** was clicked are displayed.



Note: All instances that were stopped are automatically removed from the display after *60 minutes* of being stopped. You cannot use the **Instances** page to view the history of stopped instances. You can hide the stopped instances from the display by clicking **Hide inactive**, so that you can view only the running instances.

You can find the following information about the running instances that are displayed on the **Instances** page.


- The name and path of the virtual service in the project repository are displayed in the **Name** column.
- The tags or labels added to the instance when the run was configured is displayed in the **Labels** column.




- The name of the environment associated with the operation in which the virtual service was created in Test Integrations and APIs is displayed in the **Environment** column.
- The state of the running instance is displayed as `Running` in the **State** column.
- The activity of the running instance of the virtual service based on the number of requests received in the previous hour is displayed in the **Activity** column. The activities are classified and displayed as indicated in the following table:





Activity load	Activity displayed as
Received 0 requests	None
Received 1 - 10 requests	Low
Received 11 - 100 requests	Mid
Received 101 - 1000 requests	High
Received more than 1000 requests	Very High


- The **Stop** icon  is displayed in the **Actions** column and stops the running instance of the virtual service, when clicked.

4. Complete the steps for the task that you want to perform as listed in the following table:

Task	Action
Searching for a specific instance of the virtual service for which you want to view the details or want to stop the instance.	Enter either the complete text or part of the text in the name or the path of the virtual service that is in the repository in the Search field. The instances of the virtual services that match the text searched are displayed.
Viewing the component associated with the running instance of the virtual service.	You can view the name of the associated component that is displayed on the Instances page in the Filter row only if you clicked the icon  on the System Model page to view all running instances of the virtual services .
Creating or using saved filters for viewing specific instances of virtual service resources.	To create a query, perform the following actions: <ol style="list-style-type: none"> Click New filter. Create a rule with an appropriate operator. Select criteria such as <i>Name, Path, Labels, Environment, State, Activity, or Location</i>. Select the condition, and then enter the value for the criteria. Apply the filter query. <p>The virtual services that match the filter criteria are displayed.</p> <ol style="list-style-type: none"> Save the filter query for retrieving it from the saved filters list.

Task	Action
	<p>To use a saved filter query, perform the following actions:</p> <p> Note: To open the filter query, you must have created and saved a filter query.</p> <ol style="list-style-type: none"> Click the Open filters icon . Select the saved filter. Apply the filter. <p>The virtual services that match the filter criteria are displayed.</p>
Viewing the details, configuration settings, routing rules, usage statistics, or logs of a specific running instance of the virtual service.	<p>After you identify the virtual service instance for which you want to view the details, perform the following actions:</p> <ol style="list-style-type: none"> Click the Expand icon  in the row of the identified instance of the virtual service. <p>The Details panel of the instance is displayed. You can find the following details about the instance displayed:</p> <ul style="list-style-type: none"> ▪ Name ▪ Path ▪ Labels ▪ API Environment ▪ Asset ID ▪ Git Repo ▪ Version ▪ Started at ▪ Started by ▪ Execution location ▪ Associated components ▪ Execution location <ol style="list-style-type: none"> Click Configuration to expand the panel and view the configuration details of the virtual service. You can find the following configuration details about the instance displayed: <ul style="list-style-type: none"> ▪ Dataset overrides ▪ Variables ▪ JVM args ▪ Environment variables ▪ Additional parameters Click Behavior to expand the panel and view the behavior that was configured for the virtual service.

Task	Action
	<p>d. Click Routing Rules to expand the panel and view the routing rules that were configured in the virtual service.</p> <p>e. Click Statistics to expand the panel and view the graphical representation of the activities performed by the virtual service.</p> <p>f. Click Diagnostics to expand the panel and view the logging configuration of the virtual service and the links to the logs.</p>
<p>Changing the behavior or logging level of a running instance of the virtual service.</p>	<p>After you identify the virtual service instance for which you want to view the details, perform the following actions:</p> <p>a. Click the Expand icon  in the row of the identified instance of the virtual service.</p> <p>The Details panel of the instance is displayed.</p> <p>Perform any of the following actions:</p> <ul style="list-style-type: none"> ◦ Click Behavior to expand the panel. <p>Perform the following steps to change the behavior configurations:</p> <ol style="list-style-type: none"> a. Click the Edit icon . b. Select a different operation for the virtual service, if available. c. Select a different response time and enter the values for the delay that you want to set. d. Click apply. <ul style="list-style-type: none"> ◦ Click Diagnostics to expand the panel. <p>Perform the following steps to change the logging level configuration:</p> <ol style="list-style-type: none"> a. Click the Edit icon . b. Select a different logging level. c. Click apply.
<p>Stopping a specific running instance of the virtual service.</p>	<ol style="list-style-type: none"> a. Click the Stop icon  in the row of the virtual service instance. b. Click Ok in the Stop virtual service instance dialog that is displayed. <p>The running virtual service instance is stopped.</p>

Task	Action
Stopping multiple running instances or all the running instances of virtual services.	<p>a. Perform any of the following actions:</p> <ul style="list-style-type: none"> ▪ Select the checkbox in the row of the virtual service instances that you want to stop, and then click Stop selected. ▪ Select the Select all checkbox in the header row, and then click Stop selected. <p> Note: All running instances are selected when you click the Select all checkbox.</p> <p>b. Click Ok in the Stop all selected virtual service instances dialog that is displayed.</p> <p>The running virtual service instances are stopped.</p>

Results

You have viewed running instances of the virtual services from the **Instances** page on Test Hub.

What to do next

You can perform the following tasks:

- View the configurations of a running instance of the virtual service. See [Viewing configurations of running instances of virtual services on page 775](#).
- Modify the behavior or logging level of a running instance of the virtual service before you run tests on the virtual service. See [Modifying configurations of running instances of virtual services on page 780](#).
- Stop a running instance or all running instances of the virtual services. See [Stopping virtual services on page 787](#).

Viewing configurations of running instances of virtual services

After you start virtual services from the **Resources** page, you can view the details, configuration settings, routing rules, usage statistics, or logs of a specific running virtual service instance from the **Instances** page on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Started the virtual service resources as a *Tester* or *Owner* from the **Resources** page.

About this task

After you have started the virtual services from the **Resources** page on Test Hub, you can go to the **Instances** page and view the information about a specific running virtual service instance.

1. Log in to Test Hub.


The team space that contains your project is displayed.

2. Open the project that contains the virtual service resources in the test assets by clicking **Projects > My Projects > *project_name***.

The **Overview** page is displayed.

3. Click **Virtualization > Instances** in the navigation pane.

The **Instances** page is displayed with all running instances of the virtual services that are in your project.


4. Identify the running instance of the virtual service of which you want to view the details, configuration settings, and other information.
5. Click the **Expand** icon  of the instance of the virtual service.

The **Details** panel of the instance is displayed with the following details about the virtual service instance:



Note: The information displayed pertains to the configuration settings that were configured in the virtual service at the time it was created or run.

Option	Information displayed
Name	Displays the name of the virtual service.
Path	Displays the path of the virtual service resource in the project repository.
Labels	Displays the labels or tags that were entered for the virtual service when a run of the virtual service was configured.
API Environment	Displays the name of the environment associated with the operation in which the virtual service was created in Test Integrations and APIs or the environment selected for the run.
Asset ID	The ID of the virtual service resource in the project repository that was started as an instance.
Git Repo	The git repository added to the project that contains the virtual service resources.
Version	Displays the version of the virtual service resource that is in the project repository that was started as an instance.
Started at	Displays the date and time when the virtual service instance started.
Started by	Displays the name of the project member who started the virtual service instance.


Option	Information displayed
Associated components	Displays the components in the system model that are associated with the virtual service.
Execution location	<p>Displays the location of the Kubernetes cluster on which the virtual service instance ran.</p> <p> Note: The name of the remote Kubernetes cluster is displayed if the virtual service instance ran on the remote cluster or <i>Default Cluster</i> is displayed if the virtual service ran on Test Hub.</p>


The other panels that you can view are as follows:















- Configuration
- Behavior
- Routing Rules
- Statistics
- Diagnostics


6. Expand each panel by clicking the panel to view the information in that panel.

You can view the following information in the different panels:

Panel	Details										
Configuration	<p>The Configuration panel of the instance is displayed when expanded with the following details about the virtual service instance:</p> <p> Note: The information displayed pertains to the configuration settings that were configured in the virtual service at the time it was created or run.</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Information displayed</th> </tr> </thead> <tbody> <tr> <td>Dataset overrides</td> <td>Displays the dataset overrides that were configured when executing the virtual service instance.</td> </tr> <tr> <td>Variables</td> <td>Displays the variables that were configured when executing the virtual service instance.</td> </tr> <tr> <td>JVM args</td> <td>Displays the JVM arguments that were configured when executing the virtual service instance.</td> </tr> <tr> <td>Environment variables</td> <td>Displays the environment variables that were configured when executing the virtual service instance.</td> </tr> </tbody> </table>	Option	Information displayed	Dataset overrides	Displays the dataset overrides that were configured when executing the virtual service instance.	Variables	Displays the variables that were configured when executing the virtual service instance.	JVM args	Displays the JVM arguments that were configured when executing the virtual service instance.	Environment variables	Displays the environment variables that were configured when executing the virtual service instance.
Option	Information displayed										
Dataset overrides	Displays the dataset overrides that were configured when executing the virtual service instance.										
Variables	Displays the variables that were configured when executing the virtual service instance.										
JVM args	Displays the JVM arguments that were configured when executing the virtual service instance.										
Environment variables	Displays the environment variables that were configured when executing the virtual service instance.										

Panel	Details											
	Option	Information displayed										
	Additional parameters	Displays the additional parameters that were configured when executing the virtual service instance.										
Behavior	<p>The Behavior panel of the instance is displayed when expanded with the following details about the virtual service instance:</p> <p> Note: The information displayed pertains to the configuration settings that were configured in the virtual service at the time it was created or run.</p> <table border="1" data-bbox="522 699 1421 1222"> <thead> <tr> <th data-bbox="522 699 703 753">Option</th> <th data-bbox="703 699 1421 753">Information displayed</th> </tr> </thead> <tbody> <tr> <td data-bbox="522 753 703 852">Performance</td> <td data-bbox="703 753 1421 852">Displays the optimization setting and the maximum number of threads in use by the virtual service.</td> </tr> <tr> <td data-bbox="522 852 703 993">Operation</td> <td data-bbox="703 852 1421 993">Displays the operations referenced by the virtual service. You can select each operation to view the related response time and the pass through settings.</td> </tr> <tr> <td data-bbox="522 993 703 1089">Response time</td> <td data-bbox="703 993 1421 1089">Displays the response time option that is in use for the selected operation within the virtual service.</td> </tr> <tr> <td data-bbox="522 1089 703 1222">Passthrough</td> <td data-bbox="703 1089 1421 1222">Displays the pass through option that is in use for the selected operation within the virtual service.</td> </tr> </tbody> </table>		Option	Information displayed	Performance	Displays the optimization setting and the maximum number of threads in use by the virtual service.	Operation	Displays the operations referenced by the virtual service. You can select each operation to view the related response time and the pass through settings.	Response time	Displays the response time option that is in use for the selected operation within the virtual service.	Passthrough	Displays the pass through option that is in use for the selected operation within the virtual service.
Option	Information displayed											
Performance	Displays the optimization setting and the maximum number of threads in use by the virtual service.											
Operation	Displays the operations referenced by the virtual service. You can select each operation to view the related response time and the pass through settings.											
Response time	Displays the response time option that is in use for the selected operation within the virtual service.											
Passthrough	Displays the pass through option that is in use for the selected operation within the virtual service.											
Routing Rules	<p>The Routing Rules panel of the instance is displayed when expanded.</p> <p>The routing rules that relate to the virtual service are displayed.</p> <table border="1" data-bbox="522 1394 1421 1793"> <thead> <tr> <th data-bbox="522 1394 651 1449">Column</th> <th data-bbox="651 1394 1421 1449">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="522 1449 651 1545">Activity</td> <td data-bbox="651 1449 1421 1545">Indicates the routing type.</td> </tr> <tr> <td data-bbox="522 1545 651 1677">Target</td> <td data-bbox="651 1545 1421 1677">Displays the endpoint of the target service for which requests are being routed.</td> </tr> <tr> <td data-bbox="522 1677 651 1793">Recipient</td> <td data-bbox="651 1677 1421 1793">Displays the endpoint of the virtual service that receives the routed requests</td> </tr> </tbody> </table>		Column	Description	Activity	Indicates the routing type.	Target	Displays the endpoint of the target service for which requests are being routed.	Recipient	Displays the endpoint of the virtual service that receives the routed requests		
Column	Description											
Activity	Indicates the routing type.											
Target	Displays the endpoint of the target service for which requests are being routed.											
Recipient	Displays the endpoint of the virtual service that receives the routed requests											

Panel	Details																
	<p>You can expand the panel for each intercept by clicking on the expand icon . You can then find the following information:</p> <table border="1" data-bbox="522 369 1421 1457"> <thead> <tr> <th data-bbox="522 369 662 422">Tab</th> <th data-bbox="662 369 1421 422">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="522 422 662 1314">Details</td> <td data-bbox="662 422 1421 1314"> <p>The Details panel provides the following information:</p> <table border="1" data-bbox="669 499 1414 1306"> <thead> <tr> <th data-bbox="669 499 792 552">Option</th> <th data-bbox="792 499 1414 552">Information displayed</th> </tr> </thead> <tbody> <tr> <td data-bbox="669 552 792 1014">Condition</td> <td data-bbox="792 552 1414 1014"> <p>Lists the condition that causes the traffic to be routed to a virtual service.</p> <p>You can view the detailed condition of the rule defined for the intercept by clicking the View icon .</p> <p>The detailed condition of the rule is displayed in the Full condition window.</p> <p> Note: You can copy the detailed condition if you intend to use it elsewhere.</p> </td> </tr> <tr> <td data-bbox="669 1014 792 1108">Routing to</td> <td data-bbox="792 1014 1414 1108">Lists the virtual service to which traffic is routed.</td> </tr> <tr> <td data-bbox="669 1108 792 1203">Created by</td> <td data-bbox="792 1108 1414 1203">Lists the user who created the rule.</td> </tr> <tr> <td data-bbox="669 1203 792 1306">Created at</td> <td data-bbox="792 1203 1414 1306">Lists the date and time when the rule was created.</td> </tr> </tbody> </table> </td> </tr> <tr> <td data-bbox="522 1314 662 1457">Activity</td> <td data-bbox="662 1314 1421 1457"> <p>The Activity panel lists the log of the activities performed in relation to the routing rule. This panel also displays the details of the activity, date and time, and message for that activity.</p> </td> </tr> </tbody> </table>	Tab	Description	Details	<p>The Details panel provides the following information:</p> <table border="1" data-bbox="669 499 1414 1306"> <thead> <tr> <th data-bbox="669 499 792 552">Option</th> <th data-bbox="792 499 1414 552">Information displayed</th> </tr> </thead> <tbody> <tr> <td data-bbox="669 552 792 1014">Condition</td> <td data-bbox="792 552 1414 1014"> <p>Lists the condition that causes the traffic to be routed to a virtual service.</p> <p>You can view the detailed condition of the rule defined for the intercept by clicking the View icon .</p> <p>The detailed condition of the rule is displayed in the Full condition window.</p> <p> Note: You can copy the detailed condition if you intend to use it elsewhere.</p> </td> </tr> <tr> <td data-bbox="669 1014 792 1108">Routing to</td> <td data-bbox="792 1014 1414 1108">Lists the virtual service to which traffic is routed.</td> </tr> <tr> <td data-bbox="669 1108 792 1203">Created by</td> <td data-bbox="792 1108 1414 1203">Lists the user who created the rule.</td> </tr> <tr> <td data-bbox="669 1203 792 1306">Created at</td> <td data-bbox="792 1203 1414 1306">Lists the date and time when the rule was created.</td> </tr> </tbody> </table>	Option	Information displayed	Condition	<p>Lists the condition that causes the traffic to be routed to a virtual service.</p> <p>You can view the detailed condition of the rule defined for the intercept by clicking the View icon .</p> <p>The detailed condition of the rule is displayed in the Full condition window.</p> <p> Note: You can copy the detailed condition if you intend to use it elsewhere.</p>	Routing to	Lists the virtual service to which traffic is routed.	Created by	Lists the user who created the rule.	Created at	Lists the date and time when the rule was created.	Activity	<p>The Activity panel lists the log of the activities performed in relation to the routing rule. This panel also displays the details of the activity, date and time, and message for that activity.</p>
Tab	Description																
Details	<p>The Details panel provides the following information:</p> <table border="1" data-bbox="669 499 1414 1306"> <thead> <tr> <th data-bbox="669 499 792 552">Option</th> <th data-bbox="792 499 1414 552">Information displayed</th> </tr> </thead> <tbody> <tr> <td data-bbox="669 552 792 1014">Condition</td> <td data-bbox="792 552 1414 1014"> <p>Lists the condition that causes the traffic to be routed to a virtual service.</p> <p>You can view the detailed condition of the rule defined for the intercept by clicking the View icon .</p> <p>The detailed condition of the rule is displayed in the Full condition window.</p> <p> Note: You can copy the detailed condition if you intend to use it elsewhere.</p> </td> </tr> <tr> <td data-bbox="669 1014 792 1108">Routing to</td> <td data-bbox="792 1014 1414 1108">Lists the virtual service to which traffic is routed.</td> </tr> <tr> <td data-bbox="669 1108 792 1203">Created by</td> <td data-bbox="792 1108 1414 1203">Lists the user who created the rule.</td> </tr> <tr> <td data-bbox="669 1203 792 1306">Created at</td> <td data-bbox="792 1203 1414 1306">Lists the date and time when the rule was created.</td> </tr> </tbody> </table>	Option	Information displayed	Condition	<p>Lists the condition that causes the traffic to be routed to a virtual service.</p> <p>You can view the detailed condition of the rule defined for the intercept by clicking the View icon .</p> <p>The detailed condition of the rule is displayed in the Full condition window.</p> <p> Note: You can copy the detailed condition if you intend to use it elsewhere.</p>	Routing to	Lists the virtual service to which traffic is routed.	Created by	Lists the user who created the rule.	Created at	Lists the date and time when the rule was created.						
Option	Information displayed																
Condition	<p>Lists the condition that causes the traffic to be routed to a virtual service.</p> <p>You can view the detailed condition of the rule defined for the intercept by clicking the View icon .</p> <p>The detailed condition of the rule is displayed in the Full condition window.</p> <p> Note: You can copy the detailed condition if you intend to use it elsewhere.</p>																
Routing to	Lists the virtual service to which traffic is routed.																
Created by	Lists the user who created the rule.																
Created at	Lists the date and time when the rule was created.																
Activity	<p>The Activity panel lists the log of the activities performed in relation to the routing rule. This panel also displays the details of the activity, date and time, and message for that activity.</p>																
Statistics	<p>The Statistics panel of the instance is displayed when expanded.</p> <p>The requests received by the instance of the virtual service in the previous hour are displayed as a graph. The total requests received is indicated in the Total hits field and the activity load in the previous hour are displayed in the Activity field.</p> <p>The view can be refreshed by clicking the Refresh icon .</p>																
Diagnostics	<p>The Diagnostics panel of the instance is displayed when expanded with the following details:</p>																

Panel	Details	
	Option	Description
	Logging	Displays the logging level that was configured for the virtual service.
	Execution ID	Displays the ID of execution for the virtual service instance.
	Stubs runtime ID	Displays the ID of the runtime where the virtual service instance runs. If executed on a cluster the value is that of the label execution-stubs-runtime-id on the pod that runs the virtual service instance. If executed on an API agent, the value is the ID of the engine on which the virtual service instance runs.
	Init container log	Displays the link to the log of the Init container.  Note: This logs contains messages that relate to configuring the container which is ready to run the virtual service.
	Main container log	Displays the link to the log of the main container. This log contains messages logged by the virtual service based on the selected logging level.
	Pod events log	Displays the link to the log of the pod events log. This log contains messages logged in the pod for the virtual service events.

Results

You have viewed the information about the configuration, routing rules, usage, or logs of the virtual service instance.

What to do next

You can modify the behavior or logging level configurations of the running virtual service instance before you run tests on the instance. See [Modifying configurations of running instances of virtual services on page 780](#).

Related information

[Configuring a run of a virtual service on page 750](#)

Modifying configurations of running instances of virtual services

After you start virtual services from the **Resources** page and before you run tests on the virtual service, you might want to modify the configurations for the behavior or the logging level of the running instance.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Started the virtual service resources as a *Tester* or *Owner* from the **Resources** page.

About this task

After you have started the virtual services from the **Resources** page on HCL DevOps Test Hub (Test Hub), you can go to the **Instances** page and identify the running instance of the virtual service for which you want to modify the configurations. You can modify the configurations for the behavior or logging levels of the running instance of the virtual service.

1. Log in to Test Hub.


The team space that contains your project is displayed.

2. Open the project that contains the virtual service resources in the test assets by clicking **Projects > My Projects > *project_name***.

The **Overview** page is displayed.

3. Click **Virtualization > Instances** in the navigation pane.

The **Instances** page is displayed with all running instances of the virtual services that are in your project.

4. Identify the running instance of the virtual service for which you want to modify the configurations.
5. Click the **Expand** icon  of the instance that you want to modify the configurations.

The **Details** panel of the instance is displayed.

6. Click **Behavior** to expand the panel and perform the following steps to change the configurations:

- a. Click the **Edit** icon .

The options that you can modify are enabled for editing.

- b. Select a different operation for the virtual service, if available, from the list for the **Operation** option.
- c. Select a different response time from the list for the **Response time** option, and then enter the values for the delay that you want to set.

You can modify the value by selecting a value from the following options:

Option	Description
No delay	Select this option for a response with no delay.
Minimum delay	Select this option for a delay in the response by entering the required delay in milliseconds (ms).
Uniform distribution	Select this option for a uniformly distributed response time by specifying the minimum and maximum delay in milliseconds (ms).
Gaussian distribution	Select this option for a response time with Gaussian distribution by specifying the minimum and maximum delay in milliseconds (ms).

d. Click **apply** to apply the modified settings.

7. Click **Diagnostics** to expand the panel and perform the following steps to change the logging level that is configured:

a. Click the **Edit** icon .

The **Logging level** option is enabled for editing.

b. Select a different logging level from the following options.

Option	Description
None	Specifies that the virtual service does not write log messages.
Normal	Specifies that the virtual service writes informational messages.
Debug	Specifies that the virtual service writes informational and debugging messages.

c. Click **apply** to apply the modified settings.

Results

You have viewed and modified the configurations of a running virtual service instance from the **Instances** page on Test Hub.

What to do next

You can run tests that are in an API Suite in your project, which then run on the virtual service instance. See [Configuring an API Suite or an API test run on page 486](#).

Viewing routing rules of the virtual services

When you run virtual services that are created to run on proxies or intercepts and have routing rules defined, you can view the proxies or intercepts and the routing rules from the **Intercept Rules** page.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Started the virtual service resources as a *Tester* or *Owner* from the **Resources** page.

About this task

After you run the virtual service resources that are in your project on HCL DevOps Test Hub (Test Hub), you can view the routing rules for the running instances of the virtual service resources, from the **Intercept Rules** page.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the virtual service resources in the test assets by clicking **Projects > My Projects > *project_name***.

The **Overview** page is displayed.


3. Click **Virtualization > Intercept Rules** in the navigation pane.

The **Intercept Rules** page is displayed.

The agents, proxies, or intercepts that are configured in the virtual service resources that are in your project are displayed.







You can find the following information that is displayed on the **Intercept Rules** page.

Column	Description
Activity	Lists the agent or proxy.
Target	Lists a summary view of the endpoint that was recorded or virtualized.
Recipient	Lists the destination where messages sent to are captured.

4. Expand the proxy or intercept panel by clicking the **Expand** icon .

The details of the proxy or intercept are displayed in the **Details** tab in the expanded panel.

5. Click each tab to view the following information:

Tab	Description										
Details	<p>The Details panel provides the following information:</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Condition</td> <td> <p>Lists the condition that causes the traffic to be recorded or routed to a virtual service.</p> <p>You can view the detailed condition of the rule defined for the proxy or intercept by clicking the View icon .</p> <p>The detailed condition of the rule is displayed in the Full condition window.</p> <p> Note: You can copy the detailed condition if you intend to use it elsewhere.</p> </td> </tr> <tr> <td>Routing to</td> <td>Lists the virtual service to which traffic is routed.</td> </tr> <tr> <td>Created by</td> <td>Lists the user who created the rule.</td> </tr> <tr> <td>Created at</td> <td>Lists the date and time when the rule was created.</td> </tr> </tbody> </table>	Option	Description	Condition	<p>Lists the condition that causes the traffic to be recorded or routed to a virtual service.</p> <p>You can view the detailed condition of the rule defined for the proxy or intercept by clicking the View icon .</p> <p>The detailed condition of the rule is displayed in the Full condition window.</p> <p> Note: You can copy the detailed condition if you intend to use it elsewhere.</p>	Routing to	Lists the virtual service to which traffic is routed.	Created by	Lists the user who created the rule.	Created at	Lists the date and time when the rule was created.
Option	Description										
Condition	<p>Lists the condition that causes the traffic to be recorded or routed to a virtual service.</p> <p>You can view the detailed condition of the rule defined for the proxy or intercept by clicking the View icon .</p> <p>The detailed condition of the rule is displayed in the Full condition window.</p> <p> Note: You can copy the detailed condition if you intend to use it elsewhere.</p>										
Routing to	Lists the virtual service to which traffic is routed.										
Created by	Lists the user who created the rule.										
Created at	Lists the date and time when the rule was created.										
Activity	The Activity panel lists the log of the activities performed by the proxy or intercept. This panel also displays the details of the activity, date and time, and message for that activity.										

Results

You have viewed the routing rules for the virtual service, proxy, or intercept that are running or connected to Test Hub.

What to do next

You can perform any of the following tasks:

- You can view the usage statistics of the virtual services that are running. See [Viewing usage statistics of virtual services on page 784](#).
- You can stop the running instances of the virtual services. See [Stopping virtual services on page 787](#).

Related information

[Management of virtualized services on page 741](#)

Viewing usage statistics of virtual services

After you start virtual service resources that are in the project repositories, and then run tests on the virtual service, you can view the usage statistics of the virtual services from the **Usage** page on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Completed the prerequisite tasks before you configure a run of the HTTP virtual services. See [Prerequisites for running HTTP virtual services in a Kubernetes cluster on page 742](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Verified the version of the supported assets and resources that you want to use. See [Supported versions of assets and resources on page 88](#).
- Started the virtual service resources as a *Tester* or *Owner* from the **Resources** page.
- Ran API Suites with tests that run on the virtual service resources that are configured.

About this task

After you run the virtual services that are in a project, you can view details of the usage statistics of the virtual services from the **Usage** page on Test Hub.



Restriction: The usage statistics of virtual services if they are run on Test Integrations and APIs agents, are collected only if the agents are of the same version as Test Hub.



Important: The settings such as the **Period**, or **From** and **To** that you configure to view the usage details are retained for you during your ongoing session and the settings are retained even when you perform the following activities:

- Log out of the server and log in again.
- Log in after a server restart.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Open the project that contains the virtual service resources in the test assets by clicking **Projects > My Projects > *project_name***.

The **Overview** page is displayed.

3. Click **Virtualization > Usage** in the navigation pane.

The **Virtual Service Usage** page is displayed.


The total requests that were received by all virtual services on the server are displayed as a graph.

The default view displays the usage data of the virtual services that are gathered during the previous week and grouped in an hourly interval. The total requests received from when the virtual service was started are displayed.

The metric data that is reported is for the number of requests that are received by a virtual service. The requests can be requests, messages, or other calls that are received by the virtual service.

4. Change the default settings to view the graph for the period for which you want to view the usage.

You can change the default view by selecting an option from the following options:

Option	Description
<i>1 Hour</i>	Displays the usage data of the virtual service that are gathered during the previous hour and grouped in a 20-second interval.
<i>1 Day</i>	Displays the usage data of the virtual service that are gathered during the previous day and grouped in an eight-minute interval.
<i>1 Week</i>	Displays the usage data of the virtual service that are gathered during the previous week and grouped in an hourly interval.  Note: This is the default view.
<i>Range</i>	Displays the usage data of the virtual service that are gathered during the date range specified in the From and To fields. You can select the date and time for both the From and To fields. The data is dynamically grouped in intervals that depend on the range selected.



Note: You can refresh the view at any time to present the graph for the latest data by clicking **Refresh**.

5. View details of the virtual service by hovering the mouse over any of the data-points to view the details at that point.

The following details about the virtual service are displayed when you hover the mouse over the data-point:

- The date and time of the request received.
- The number of requests received.

Results

You have viewed the usage details of the virtual services in your project that ran on Test Hub.

What to do next

You can perform any of the following tasks:

- You can view the routing rules defined in the virtual service, agent, proxy, or intercept. See [Viewing routing rules of the virtual services on page 783](#).
- You can stop the virtual service instances that are running. See [Stopping virtual services on page 787](#).

Related information

[Management of virtualized services on page 741](#)

Stopping virtual services

After you started instances of the virtual services from the **Resources** page on HCL DevOps Test Hub (Test Hub), you must stop the running instance from the **Instances** page when you do not want to use the instance. You can stop a single instance, multiple, or all running instances.

Before you begin

You must have started the virtual service from the **Resources** page. See [Configuring a run of a virtual service on page 750](#).

1. Log in to Test Hub.


The team space that contains your project is displayed.

2. Open the project that contains the virtual service resources in the test assets by clicking **Projects > My Projects > *project_name***.


The **Overview** page is displayed.


3. Go to the **Instances** page and identify the instance of the virtual service that you want to stop.

You can select the virtual service instances that satisfy the following conditions:

- The virtual service instance must be either in the **In transition** or **Running** state.
- The **Stop** icon  in the **Actions** column must be enabled for the virtual service instance.
- The checkbox icon must be enabled for selection for the virtual service instance.

4. Perform the actions indicated in the following table:

If...	Action...
You want to stop any virtual service instance that is running	Perform any of the following actions: <ul style="list-style-type: none"> ◦ Click the Stop icon  in the Actions column of the virtual service instance. The Stop virtual service instance dialog is displayed.

If...	Action...
	<ul style="list-style-type: none"> ◦ Select the checkbox in the row of the virtual service instance, and then click Stop selected. <p>The Stop all selected virtual service instances dialog is displayed.</p>
You want to stop multiple virtual service instances that are running	<p>Perform any of the following actions:</p> <ul style="list-style-type: none"> ◦ Select the checkbox in the row of the virtual service instance, and then click Stop selected. ◦ Select the Select all checkbox in the header row, and then click Stop selected. <p> Note: All running instances are selected when you select the Select all checkbox.</p> <p>The Stop all selected virtual service instances dialog is displayed.</p>

5. Click **Ok**.

A notification is displayed that the running virtual service instances are stopped.

Results

You have stopped a single instance, multiple, or all running virtual service instances. The stopped virtual service instance on the **Instances** page is displayed with the status as *Stopped by User* in the **State** column.

What to do next

You can restart the virtual service by completing the configurations that you want from the **Resources** page.

You can open and view the `Main container log` of the stopped virtual service.

Test results

After the tests, Suites, or schedules are run and completed, you can view the results and open the reports in HCL DevOps Test Hub (Test Hub) to analyze the verdict, the performance, and statistics. You can also re-execute tests and schedules from the **Results** page with the same commit id.

Summary

The **Results** page displays the verdict summary for all the tests run in the project.

In this page, you can perform any of the following tasks:

- Display the suites only, or the tests only, or tests and suites that are run with advanced settings.
- Searching for test results. You can filter test results by using any of the following ways:

- By using the **Search** field to search for results by name.
- By selecting predefined time interval from the **Selection interval** list.
- By selecting **To** and **From date** from the **Date Interval** option.
- By clicking verdict from the **Verdict summary** slider (*Pass, Fail, Inconclusive, or Error*).
- By creating filter queries.
- Adding labels if you have a tester or an owner role.
- Adding text in the **Notes** field. You can also group the notes similar to a conversation in a chat channel.
- Locking test results: Locked results can be unlocked by the project owner or the project member who locked the results.
- Deleting test results.
- Comparing performance reports.
- Viewing trending reports.
- Viewing multiple reports depending on the test types and the parameters set when you execute the tests: Statistics reports, Functional Test reports, Test logs or Result data queries
- Re-executing tests with the same commit id.

For more details, see the links in the next section of this page.

Test results and reports

HCL DevOps Test Hub (Test Hub) is a single location for hosting the results and reports of all tests run on different desktop clients and for tests run from the server.

Test results list

When a test run is complete, you can see the results on the **Results** page that you open from **Analyze > Results** in the navigation pane.

The list of test results is organized by types, you can display the suites only in the **Suites** tab, or the tests only in the **Tests** tab, or the tests and suites that are run with advanced settings in the **Advanced** tab.

The screenshot shows the Test Hub Results page interface. At the top, there is a search bar on the left and a 'Selection interval' dropdown set to 'Custom' on the right. Below this is a 'Verdict summary' bar with a green segment for 'Pass 50.0%' and a red segment for 'Fail 50.0%'. A 'New filter +' button is visible on the left. The main content is a table with the following columns: Type, Test, Started by, Started on, Duration, Verdict, Components, Labels, and Actions.

Type	Test	Started by	Started on	Duration	Verdict	Components	Labels	Actions
<input type="checkbox"/>	MultiplySuite StubTestNew/Logical/calcul...	meenal123	Apr 28, 2021 4:49 PM	00:00:23.721	Pass	serviceA UICompOTS		+ ▶ ⋮
<input type="checkbox"/>	AddSuite StubTestNew/Logical/calcul...	meenal123	Apr 28, 2021 4:42 PM	00:00:16.312	Fail	ServiceAB UICompOTS		+ ▶ ⋮

Results, reports, and logs are generated for tests run from Test Hub, or from desktop clients such as Test Performance, Test UI, or Test Integrations and APIs.

You must have configured the desktop clients to publish reports of tests that are run from the desktop client to the Test Hub. For more information about the publishing procedure, refer to the links at the end of this page.

When you click the test result, a results panel is displayed with information in the following tabs:

- Details
- Reports
- History

Test result details

You can view the following details about the test assets used in the test run in the **Details** tab:

- The status of the test results, the date and time it was executed and completed.
- The details about the Git repository that contains the test assets.

aries-web_transactions



Status: Complete

an hour ago finished

Details

Report

History

master branch

Description

GIT details

Asset id e46687c6-8d53-4fec-a687-c68d537fec5d

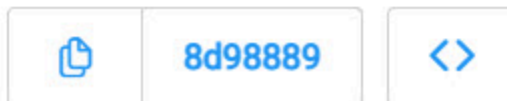
Repo git@github01.hclpnp.com:testingproducts/FT_SVT_T
estAsset.git

Path IOS_Vatsala_WebUIProject/Tests/aries-web_transactions.t
estsuite

Commit • **Large Report testing 1053 - Web UI and Compound
Test Loop 10 and 11.**

COCOU

- anshul.kulkarni committed 3 days ago
- (#HEAD) (#refs/heads/master)



You can configure the information displayed in the **Details** tab and the **Results** columns from the **View Settings** window. You access this window by clicking the **Settings** icon .

Depending on the attribute you cleared in the **View Settings** window, the **Details** card displays additional information about the test results.

The selected attributes are displayed in columns of the Result view.

The screenshot displays the HCL DevOps Test Hub interface. At the top, there is a 'New filter +' button and a settings icon. Below this is a table with columns: Id, Type, Test, Branch, Version, Result, Started by, and Started on. Two test runs are listed: ID 1004 and ID 1005. The details for test ID 1005 are expanded, showing a 'DETAILS' section with fields like Finish on, Duration, Status (Complete), Verdict (Pass), Labels, Git repo, Git path, Git commit, and Execution location. To the right of the details is a 'Reports' section with a list of generated reports: Statistics Report, Mobile and Web UI Report, Functional Test Report, and Test Log, each with a trash icon.

Id	Type	Test	Branch	Version	Result	Started by	Started on
1004		emptyCm... WebUIProje...	master	b983bfe commit: b9...	emptyCm...	nm	Jun 2, 2020 4:59 PM
1005		cmptest-ja... WebUIProje...	master	b983bfe commit: b9...	cmptest-ja...	nm	Jun 2, 2020 5:07 PM

DETAILS

Finish on: Jun 2, 2020 5:09 PM

Duration: 00:02:25.608

Status: ●—●—●—○
Complete

Verdict: ■ Pass

Labels: +

Git repo: git@github01.hclnp.com

Git path: WebUIProjectCustomCode/Compound...

Git commit: b983bfe
commit: b983bfe29573dc69b514a5596b

Execution location: Default Cluster




Reports



- Statistics Report 🗑️
- Mobile and Web UI Report 🗑️
- Functional Test Report 🗑️
- Test Log 🗑️

Reports


The **Report** tab provides the following information:


- The location where the test was run.
- The reports that are generated for the test run.
- The labels that you created for the test during its configuration for a run.
- Notes that you can add to the specific test run result or displays the notes already added by you or others.








aries-web_transactions   


Status:  Complete
 an hour ago finished


Details **Report** History


 **Execution location**
 Default Cluster

 **Reports**


Statistics Report	 
Functional Test Report	 
Test Log	 
Queries	

 **Labels**



 **Notes**

You can perform the following actions on the reports and logs that are displayed on the **Reports** tab:

- View the report that opens in a web browser by clicking the report.
- Download the report by clicking the **Download** icon . You can enter a name and select the location on your computer where you want to save the report.

The report is downloaded in any of the following formats:

- A compressed file and contains the HTML pages for the supported tests.
- A compressed file that contains the Surefire report.

- A PDF for the supported tests.



Note: For tests that contain more than 30 test steps, the details of the test steps are not included in the PDF.

- Delete the report by clicking the **Delete** icon .



Note: The deleted report is not available on the server. You might have to run the test again to generate the report.

As a default configuration, test logs are delivered in a traditional format for the executed Compound Tests and schedules.

The reports are generated for Compound Tests, Schedules, and Test Suites, or for other tests that are run from a desktop client or from Test Hub.

The following table lists the reports that are generated for each type of test and the links to help pages.

Table 47. Analyzing Reports

Report	Tests type	Product	More information
Functional Test Report	API Suite	Test Integrations and APIs	Viewing reports
	Compound Test	Test UI	Unified Report
	AFT Suite		
	Functional test (For Web UI tests)		
Mobile and Web UI Report	Compound Test	Test UI and Test Performance	Mobile and web UI test result reports
	AFT Suite	Test UI	
Statistics Report	Compound Test	Test UI and Test Performance	Reports and counters. Service Performance report. From the Statistic reports, you can add counters. Refer to Adding additional counters on a separate page.
	AFT Suite	Test UI	
	VU Schedule	Test Performance	
	Rate Schedule		

Table 47. Analyzing Reports (continued)

Report	Tests type	Product	More information
	Functional test (For Web UI tests)		You can also display the statistics of counters on graphs. Refer to Displaying counter data in tables or as graphs..
Test Log	Compound Test	Test UI and Test Performance	
	AFT Suite	Test UI	
	API Suite	Test Integrations and APIs	
	Functional Test (Web UI)	Test UI	
	VU Schedule	Test Performance	
	Rate Schedule		

Creating search queries to filter test results

To filter the test and suite results list, you can create a search query that you apply as a filter in HCL DevOps Test Hub (Test Hub).

About this task

To filter the list of test and suite results, you can use rules and group of rules in your query. This procedure explains how to create a query that contains two rules and a group of rules as an example, and how to select other filters.

1. Click **New filter** on the Result page.

Follow these steps to add a rule in your query:

2. Select **Add a rule** in your query, and enter or select the required attribute in the fields to define the rule:

- a. Select the first attribute that you want to query in the drop-down list.

Example

Select **Verdict**.

- b. Select the condition as *equal to (=)* or *not equal to (!=)*, *Contain* or *Not contain*.

Example

Select =.

- c. Select a value for the first attribute you selected.

The content of this field depends on the option you selected as first attribute. Enter a value for the **id**, **TypeTest**, **Version**, **Result**, **Labels**, **Branch**, **Test path**, and **Text** attributes. For the other attributes, select a value in the drop-down list.

Example

Select **Fail**.

d. Select either the **AND** or the **OR** operator in the first field to add a condition to compare the first rule with another rule.

e. Select **Add rule** in the drop-down list.

f. Select an attribute, a condition, and select or enter a value.

Example

Select **Test** as an attribute, select = as a condition operator, and enter *mytest* as a value.

Follow these steps to add a group in your query:

3. Select **Add group** in the drop-down list.



Note: A group is a collection of rules and can also contain other groups within a query. You can add a rule or a group within this group.

4. Select a condition operator.

Example

Select **Or**.

5. Select **Add rule** in the drop-down list to add a rule in the group.

6. Select the required an attribute, an operator, and enter a value for the rule.

Example

Select the **Type** attribute, = as an operator, and **Compound Test** as a value.

7. Select **Add rule** to add another rule in the group.

8. Select and enter the appropriate parameters as you did for the other rules.

Example

Select **Type** as an attribute, = as a condition, and **VU Schedule** as a value.

9. Click **Apply** if you want to apply the filter now.

You can cancel the query or save the query when the results are found.

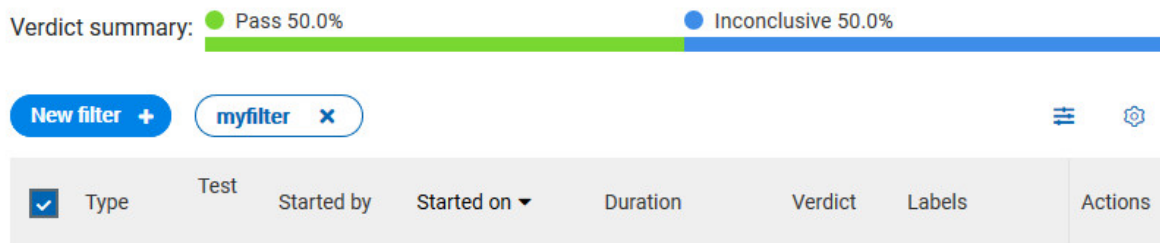
10. Click **Save**.

Proceed as follows to select your filters:

11. Enter a name for the filter and save in the dialog that is displayed.

Result

The filter is displayed as a favorite filter.



12. Click the icon to select the filters you created from the list of saved filters.

Filters

Search for filter

- None
- myfilter

Delete

Apply

- Click a filter in the list of saved filters. If a filter does not display in the list, enter the name of a filter in the search field and select it.



Note: You can delete a filter from the list or disable filters by selecting **none** in the list.

- Apply the selected filter.

Result


The filter is displayed as a new favorite filter in the **Results** page.

Comparing test reports

After you run performance tests, you can analyze the difference between two or multiple test results in HCL DevOps Test Hub (Test Hub). For example, you can compare the performance of an application at different time slots or different milestone builds between two test results.

About this task

You can compare the test runs for performance tests that are in the same project or in different projects.

- Select results in the **Results** page by checking the boxes in the first column.
- Click the  icon



Note: When comparing multiple runs, you cannot compare multiple time-ranges or stages.



Result


The two or multiple test results are displayed in a browser window in the same report.

Viewing trending reports

When the performance test runs are complete, you can view the trend of response time for an application over a period of time in HCL DevOps Test Hub (Test Hub) from trending reports. In addition to the response time, you can view the trend for the loops, transactions, and performance requirements for the application.

About this task

The trending reports are available for performance tests only.

1. Select results in the first column in the **Results** page.
2. Click the  icon to view the trending reports.



Note: When comparing multiple runs, you cannot compare multiple time-ranges or stages.



Results

You can analyze the trend information for multiple test results in the same report from a browser window.

Re-executing tests from results

You can re-execute a test, a schedule, a collection of tests or schedules from HCL DevOps Test Hub (Test Hub) if you want to retain the same commit id.

Before you begin

You must have executed a test or multiple tests from the **Execution** page in Test Hub.

Re-executing a test from the **Results** view requires a Tester or an Owner role.

About this task

The following procedure describes how to re-execute a single test or multiple tests at a time.

Perform all the following steps from the Results page.

- Proceed as follows to re-execute one test at a time:

1. Identify the test that you want to re-execute and click the **Re-execute** button in the Actions column.

Result

The **Execute test asset** dialog displays the same parameters as the ones that were set to execute the initial test from the **Execution** page.

2. Select another **Version** of the test.




Note: If you select another version of the test, the settings configured in the dialog and in the **Environment**, **Data Sources** and **Locations** tabs might change.

3. Modify the **Schedule** parameters.



Note: If you don't want to execute the test now, schedule the time when you want the test to be re-executed.


4. Click the **Environment** tab and select another **API test environment** if multiple environments were initially set for a test that is running an API Suite.
5. Click the **Data Sources** tab and select an override option if multiple data sources were defined for the initial test run.
6. Click the **Variables** tab and add a new variable if you want to re-execute a test asset with a different variable from the one configured in the asset.
7. Click the **Location** tab to override the default cluster location if the test asset has a docker host or static agents configured.
8. Click **Advanced** to modify the advanced parameters that were set for the initial test run.
9. Enter a new instance of **Java arguments** if applicable to the test.
10. Enter the environment variables that must be passed to the test run at runtime in the **Environment Variables** field.
11. In the **Resource Monitoring** tab, click the , and press the Ctrl + Space keys, or enter the initial letter of a label to select a label in the list and override the current sources in the schedule.



Note: The **Resource Monitoring** tab is displayed for performance schedules and only if you have enabled the **Resource Monitoring from Service** option from Test Performance.

The label field is grayed if you don't have any label entered in your current project. To collect data during the test execution, you must first enter labels in the Resource Monitoring sources page from Test Hub.

For more details about controlling Resource Monitoring sources and labels, see the links at the end of this page.

12. Click the **Re-execute**  icon to re-execute the test now or at a scheduled time.

- Proceed as follows to re-execute multiple tests at a time:

1. Click the checkboxes for the tests that you want to re-execute.

You can select them one by one or select all of them.

Result

A toolbar is displayed.

2. Click the **Re-execute** icon  in the toolbar.

Result

A message indicates the number of selected tests and the number of tests to be re-executed. The tests are re-executed with the same initial commit ids and parameters.

3. Click **Execute**.

Result

You can view the results, logs, and reports of the re-executed tests in the **Results** page.

Related information

[Adding a project on page 1013](#)

[Test results and reports on page 789](#)

[Controlling resource monitoring sources in a schedule on page 878](#)

[Test run configurations on page 446](#)

Test results data in Grafana

By using HCL DevOps Test Hub (Test Hub) Results Data and Grafana, you can retrieve and explore test log data that is generated in Test Hub. You must install and configure the Test Hub Results Data plugin in Grafana and create an instance of a data source that you use to explore your test log data.

Grafana is a visualization and analytics software. You can use Grafana to explore Test Hub metrics.

You must use Test Hub Results Data, which is a data source plugin, to make requests to Test Hub and represent data in a Grafana dashboard.

Installing and configuring the DevOps Test Hub Results Data plugin

You must install and configure the DevOps Test Hub Results Data plugin that is packaged along with HCL DevOps Test Hub (Test Hub) to view and analyze the log data in Grafana. You can view the test log data that is generated when you run a test from Test Hub in a Grafana dashboard.

Before you begin

You must have installed Grafana on your computer.

1. Perform the following steps to install the DevOps Test Hub Results Data plugin:
 - a. Download the DevOps Test Hub Results Data plugin from the following URL:

`<your server address>/artifacts/grafana/devops-testhub-datasource.zip`

- b. Install the DevOps Test Hub Results Data plugin by extracting the compressed file into your Grafana plugin directory.
 - c. Restart the Grafana server.

2. Perform the following steps to configure the data source by using the DevOps Test Hub Results Data plugin:
 - a. Enter the address of the local Grafana server in your web browser:



Important: You must sign in with `http://localhost:3000` or `http://localhost:8080` URLs to use the plugin.

- b. Follow these steps to verify that the DevOps Test Hub Results Data plugin is available in Grafana:
 - i. Click **Administration > Plugins and data > Plugins** in the left navigation panel of Grafana.
 - ii. Enter **DevOps Test Hub Results Data** in the search bar in the **Plugins** tab.
 - iii. Click **DevOps Test Hub Results Data** in the search results.




The DevOps Test Hub Results Data page is displayed. You can find the links to Test Hub documentation and the description of the plugin on this page.


- c. Perform the actions indicated in the following table to create an instance of a data source in Grafana by using the DevOps Test Hub Results Data plugin on the **DevOps Test Hub Results Data** page:



Note: Users with the role of a server administrator can add data sources.

Options	Actions
If you want to create an instance of the data source in Grafana by	Perform the following steps: <ol style="list-style-type: none"> i. Click Add new data source. The data source configuration panel is displayed.

Options	Actions
<p>using the proxy connection</p>	<ul style="list-style-type: none"> ii. Click Default if you want to set the data source as the default data source in Grafana. iii. Enter a name to identify the data source. iv. Enter the URL of Test Hub in the Server field. v. Enter the token that you created in Test Hub. <p> Note: The Use proxy is enabled by default.</p> <ul style="list-style-type: none"> vi. Click Save and test. <p> Note: If the server certificate is self-signed, an error message is displayed. You must add the CA certificate to establish the connection with the server.</p> <ul style="list-style-type: none"> vii. Perform the following steps to add the CA certificate: <ul style="list-style-type: none"> 1. Enable With CA Cert by using the toggle icon. 2. Add the CA certificate in the CA Cert field that you saved during the installation of Test Hub. 3. Click Save and test.
<p>If you want to create an instance of the data source in Grafana with additional Helm parameters</p> <p> Important: Before you install Test Hub, you must use an additional Helm command parameter router.allowedOrigin during the installation. You can also run this command after installation. See the</p>	<p>Perform the following steps:</p> <ul style="list-style-type: none"> i. Click Add new data source. <p>The data source configuration panel is displayed.</p> <ul style="list-style-type: none"> ii. Click Default if you want to set the data source as the default data source in Grafana. iii. Enter a name to identify the data source. iv. Enter the URL of Test Hub in the Server field. v. Enter the token that you created in Test Hub. vi. Disable Use proxy by using the toggle icon. vii. Click Save and test.

Options	Actions
 Additional Helm parameters topic in the related information section.	

Results

You have installed and configured the DevOps Test Hub Results Data plugin in Grafana.

What to do next

You can retrieve the test log data from Test Hub by using queries.

Related information

[Additional Helm parameters on page 111](#)

[Download Grafana](#)

[Install Grafana plugins](#)

[Managing access to DevOps Test Hub on page 994](#)

[Add a data source](#)

[Explore](#)

[Viewing test results data in Grafana on page 804](#)

Viewing test results data in Grafana

After you install and configure DevOps Test Hub Results Data plugin in Grafana, you can retrieve the test log data that is generated in HCL DevOps Test Hub (Test Hub) and view the test results data in Grafana. You can use queries to display graphs, statistics, or tables. You can also view the details in a Grafana dashboard.

Before you begin



You must have completed the following tasks:

- Ran tests to get test logs results in Test Hub. For more information about the procedure, refer to [Test logs on page 808](#).
- Installed the DevOps Test Hub Results Data plugin and created an instance of data source for DevOps Test Hub Results Data. See [Installing and configuring the DevOps Test Hub Results Data plugin on page 802](#).
- Opened Grafana with one of these ports: `http://localhost:3000` or `http://localhost:8080`.



1. Click **Explore** in the **Toggle** menu in Grafana.

The **Data query** page with the default data source is displayed.

2. Select **DevOps Test Hub Results Data** from the list of data sources.
3. Select the type of query by performing any of the following actions when you want to view statistics, events, or activities from the test data:


To view	Actions
The variation of a metric over time.	Perform the following steps: <ol style="list-style-type: none"> a. Click the time-series statistics tab icon . b. Go to Step 4 on page 805.
The values of the metric computed over the entire time range.	Perform the following steps: <ol style="list-style-type: none"> a. Click the overall statistics tab icon . b. Go to Step 4 on page 805.
The events recorded in the test log.	Perform the following steps: <ol style="list-style-type: none"> a. Click the Events tab. b. Go to Step 4 on page 805.
The activities recorded for all events in the test log.	Perform the following steps: <ol style="list-style-type: none"> a. Click the Activities tab. b. Go to Step 4 on page 805.

4. Define the scope for the query by performing the actions indicated in the following table:


Options	Actions
If you want to use the data from the team space	 Note: You can select the team space as the scope for your query only if you want to view the time-series statistics or overall statistics. Perform the following steps: <ol style="list-style-type: none"> a. Select the Team Space option. b. Select the team space from the list of team spaces that are listed. c. Select either Service, if you want to use Resource Monitoring data or Stubs, if you want to use the virtual service data in the query. d. Go to Step 5 on page 806.
If you want to use the data from your project	 Note: You can select the project as the scope for your query if you want to view any of the supported query types. Perform the following steps: <ol style="list-style-type: none"> a. Select the Project option. b. Select the project from the list of projects that are listed.

Options	Actions
	<p>c. Select the Result option.</p> <p>d. Select the result from the list of results that are displayed. Alternatively, you can also enter text for the name or ID of the result in the Search field, and then select the result from the listed items.</p> <p>e. Go to Step 5 on page 806, if you selected the query type as time-series statistics or overall statistics, or go to Step 6 on page 806, if you selected the query type as Events or Activities.</p>

5. Perform the following steps to select metrics for your query:

- a. Select a metric from the list of metrics in the row of the **Metrics** option.
- b. Assign a name for the metric that helps you identify it, by entering text in the **Metric** field.
- c. Select the aggregation factor for the metric. For example, select **Sum** to add all values of the metric.
- d. Optionally, click the  icon to add a separator.

With separators, you can view details of a specific project or test result in comparison with the values of the metric across projects or results in the team space or project.

- e. Optionally, click the **Cumulated** toggle-button if the values for the metric must be added cumulatively over time. The default option is set to be non-cumulative.
- f. Add another metric by clicking the **Add metric** icon .



Note: If you added multiple metrics and you want to remove a metric, you can click the

Remove metric icon  of the metric.

- g. Go to [Step 7 on page 807](#).

6. Perform the following steps for the **Events** or **Activities** query type:

- a. Enter the **Query Filters**.
- b. Enter the **Timing value** and **Value**.

The timing values are sorted in ascending order. The value must be numeric so that it can be used in a graph.



Tip: To find all the information needed on the **Explore** page of DevOps Test Hub Results Data, perform any of the following actions:



- Select a result on the **Queries** page in Test Hub and copy the result number, query filters for the events or the activities, query filters, and timing values. You can find the project number in the URL of the **Queries** page.
- Click **Copy curl command** on the **Queries** page in Test Hub and paste the curl command in a text editor. You can find all the information you need in the following command.

```
curl -X 'POST' -H 'accept: application/json' -H 'Content-Type: application/json' -d '{"operator":"and","rules":[{"field":"type","operator":"is","value":"JMeterAction"}]}' https://[redacted]
https://[redacted]com/rest/projects/85350/results/1171/data/views/table/elements/?fields=creationEvent.time,duration&sort=creationEvent.time.asc'
```

7. Click **Add query** if you want to combine queries from different projects, results ids, or with different properties in the query filters.
8. Click **Run query**.

Results

You configured a query to get the test log data from Test Hub, and you can view and analyze the log data displayed for your query.

What to do next

You can build a dashboard including graphs, charts, gauges, tables to get customized visual representations of your test result data and select options to display the test result values. See [Dashboard overview](#) for more details about dashboards in Grafana.

Viewing Surefire format reports

After you execute JMeter, JUnit, API, and Functional tests with advanced settings in HCL DevOps Test Hub (Test Hub), you can view the test results in a Surefire format report by using dedicated URLs.

Before you begin

You must have configured your tests with one of the following parameters:

- For JMeter, JUnit, Postman, and HCL® AppScan Suites, set the **history** parameter with **cisterna** value in the **Additional Configuration Parameters** field in the **Advanced** settings.
- For API Suites, AFT Suites, Compound Tests, Functional tests, and API tests, set the **-history cisterna** parameter in the **Program Arguments** field in the **Advanced** settings.

About this task

When a test run is complete, you can see the test result data in a report that is in a Surefire format. You can view this report by using dedicated URLs.

1. Click **Analyze > Results** in the navigation pane.

Result

The **Results** page is displayed.

2. Click a test result, and note its project identifier and test result identifier (ID) in the URL of the test log that is displayed in your browser. The URL is in the following format: `https://myProductionServer.com/#/Initial/project/<ProjectID>/analyze/test-log/<ResultID>`.

The identifiers are required to view the Surefire test logs in the following steps.

3. Enter one of the following URLs with the appropriate information in a web browser or in an external tool that you use to view Surefire reports:

- `/rest/projects/<ProjectID>/data/views/surefire/`

Enter your REST API address and the project identifier to view the list of all test cases in the project with details about their status.

Example: `https://myProductionServer.com/rest/projects/230200/views/surefire/`

- `/rest/projects/<ProjectID>/results/<ResultID>/data/views/surefire/`

Enter your REST API address, the project identifier, and the test result identifier to view the results for this test result id in the project.

Example: `https://myProductionServer.com/rest/projects/230200/results/80250/data/views/surefire/`

Results

The test result reports are displayed in a Surefire format.

Related information

[Test logs on page 808](#)

[Test log queries on page 823](#)

Test logs

When you start a run of a test asset, HCL DevOps Test Hub (Test Hub) starts to log information about the test run in a test log. You can view the test log while the test runs and after the run is completed from the **Test Log** page.

Overview


Test Hub captures the test log for the supported tests that you can run on Test Hub. For the list of supported tests, see [Suites and tests that are supported on DevOps Test Hub on page 89](#).

Test Hub uses a built-in PostgreSQL database to store information about the test while the test runs on the server. Test Hub captures and displays the test log as the default configuration and you do not have to configure other settings during or after installation of the server.

Supported formats of test logs

The formats in which you can view the test logs are as follows:

Table 48. Supported formats for test log display

Log format	Description	Action
Default format	<p>Displays the test log in the built-in format on the Test log page. For example, the test log can be as follows:</p> 	You do not have to set any Program argument while you configure the test run.
Text format	<p>Displays the test log in a plain text format and is displayed in a different browser tab or window than the one that displays the Test log page. For example, the test log can be as follows:</p> <pre> ----- Name: autoResults Description: Test Version: null Verdict: fail Type: com.ibm.rational.test.lt.lttest ----- Name: DefaultSchedule Description: Timestamp: May 25, 2022 at 9:24:10 AM GMT null Event type: com.ibm.rational.test.lt.scheduleStart Properties: Test Log Filter (ALL): 100 (ALL) Test Log Filter (WARNINGS): 100 (ALL) Test Log Filter (ERRORS): 100 (ALL) Type: start ----- Name: DefaultUserGroup Description: Timestamp: May 25, 2022 at 9:24:10 AM GMT null Event type: com.ibm.rational.test.lt.usrGroupStart Properties: Type: start ----- </pre>	You must set the -history testlog Program Argument in the Advanced settings dialog while you configure the test run.

Activities and events

The information about the tests that is collated by Test Hub can be classified into the following types:

Table 49. Types of information collated for test logs

Information type	Description
Activity	<p>Specifies the events that are performed when the test runs as part of a test step. An activity comprises different events and is represented as a collection of events that were performed over time.</p> <p>For example, opening a specific web page by using a web browser in the test can be an activity that comprises the different events such as connecting to the web site, sending of a request, and receiving a reply. Each of these events might take a specific amount of time, which are captured in the test log.</p>
Event	<p>Specifies an action performed as part of a test step.</p> <p>For example, start connection can be an event in a test that requires a specific web page to be opened in a web browser, as detailed in the example for an activity. The exact time at which this event occurred when the test ran is captured in the test log.</p>

Test log tasks

You can find information about the tasks that you can perform on the the test logs in the **Test log** page. The test logs for the tests that you ran on Test Hub are displayed in the **Test log** page.

The tasks that you can perform on the test logs are summarized in the following table:

Table 50. Test log tasks

Task	More information
Viewing test logs after you start a test run on the Execution page.	Viewing test logs after you start a test run on page 810
Viewing test logs after the test run is completed.	Viewing test logs after the test run is completed on page 811
Viewing details about the test log.	Test log details on page 815

Viewing test logs after you start a test run

When you configure and start a test run, you can view the test log that is created progressively while the test runs.

Before you begin


You must have configured and started a test run. See [Test run configurations on page 446](#).



Note: You must have opened the **Execution** page to start the test run.

1. Click **Show in progress page** icon  in the **Actions** column of the test asset on the **Execution** page.

The asset is displayed on the **Progress** page.

2. Click the **Open action menu** icon .
3. Click the **View test log** option.

The **Test log** page is displayed with the test log collated for test run.



Note: Because the test log is collected progressively as the test runs, you must click **Refresh** to view the latest or the updated test log.

Results

You have viewed the test log for the test asset that you ran.

What to do next

You can view the details of the test log and decide on the actions you might want to take on the test log. See [Test log details on page 815](#).

Viewing test logs after the test run is completed

After you configure a test run and the run is completed, you can view the test log that is created for the test.

Before you begin

You must have configured and started a test run, and noted that the test run has the status as *Complete* displayed on the **Progress** page.

About this task


You can view the test log that is created for a specific test run from the following pages:

- The **Progress** page. To view the test log from the **Progress** page, go to [Step 1 on page 812](#).
- The **Results** page. To view the test log from the **Results** page, go to [Step 2 on page 812](#).
- The **Test log** page. To view the test log from the **Test log** page, go to [Step 3 on page 815](#).

1. Perform the following steps after the test completes its run:

- a. Go to **Execute > Progress**, if you are not already on the **Progress** page.

The test assets that are running or have completed their run are listed. If you are already on the **Progress** page, only the test asset that you selected on the **Execution** page is displayed with the status as *Complete*.

- b. Click the **Open action menu** icon  under the **Actions** column in the row of the test asset.
- c. Click the **View test log** option.

The **Test log** page is displayed with the test log collated for test run.

- d. Go to [Results on page 815](#).

2. Perform the following steps after the test completes its run:

- a. Go to **Analyze > Results**.

The test assets that have completed their run are listed.


- b. Select the tab based on the type of test assets that you ran as indicated in the following table:


If...	Then...
You want to run AFT Suites, API Suites, Compound Tests, Rate Schedules, VU Schedules, JMeter tests, JUnit tests, or Postman tests..	Click the SUITES tab.
You ran API tests, Functional tests, or Performance tests.	Click the TESTS tab.
You want to find Suites and tests by using custom filters, when you ran Suites and tests.	Click the ADVANCED tab.








































- c. Identify the test asset that you ran by performing any of the following actions:


- Scroll through the list.

 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.


 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:



Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test asset by entering any text contained in the test asset name in the **Search** field.
- Click the **Filter** icon  in the **SUITES** or **TESTS** tab to filter the displayed assets based on the asset type.

For example, select **API Suite** in the **SUITES** tab to display all API Suites or select **Functional Test** in the **TESTS** tab to display all Functional tests that are in the branch.

- Click the **Filter** icon  in the **ADVANCED** tab, and then create a filter query by using the **New filter** option by performing the following steps:
 - i. Click **New filter**.
 - ii. Enter a name for the filter.
 - iii. Select an operator, and then add a rule, or a group of rules.
 - iv. Add or enter the relevant parameters and either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- Type
 - Test Asset Name
 - Test Asset Path
 - Last Result
 - Next Run
 - Components
- v. Save the filter query to save and apply the filter query to filter the assets based on the query.

The test assets that match the filter criteria are displayed.

- Retrieve and apply a saved filter query, if you have saved filter queries previously by performing the following steps:



Note: The filter query applied previously is selected and the assets based on that filter query are displayed. To apply a different filter query, you must have created and saved the filter query.

- i. Click the **Filter** icon  in the **ADVANCED** tab.

The filter queries that you created and saved are displayed.

- ii. Click the filter that you want to apply.

The test assets that match the filter criteria are displayed.

You have identified the test asset that you ran.

- d. Click the test asset under the **Test** column.

The results and reports panel is displayed.

- e. Click the **Reports** tab, if it is not already open.

- f. Click the **Test Log** link.

The test log created for the test run is displayed in the **Test log** page.

g. Go to [Results on page 815](#).

3. Perform the following steps after the test completes its run:

a. Go to **Analyze > Test Log**.

The **Test log** page is displayed. The test assets that were run are displayed.

b. Identify the test asset by its name or the type for which you want to view the test log.

c. Click the test asset under the **Test** column.

The test log created for the test run is displayed.

d. Go to [Results on page 815](#).

Results

You have viewed the test log for the test asset that you ran.

What to do next

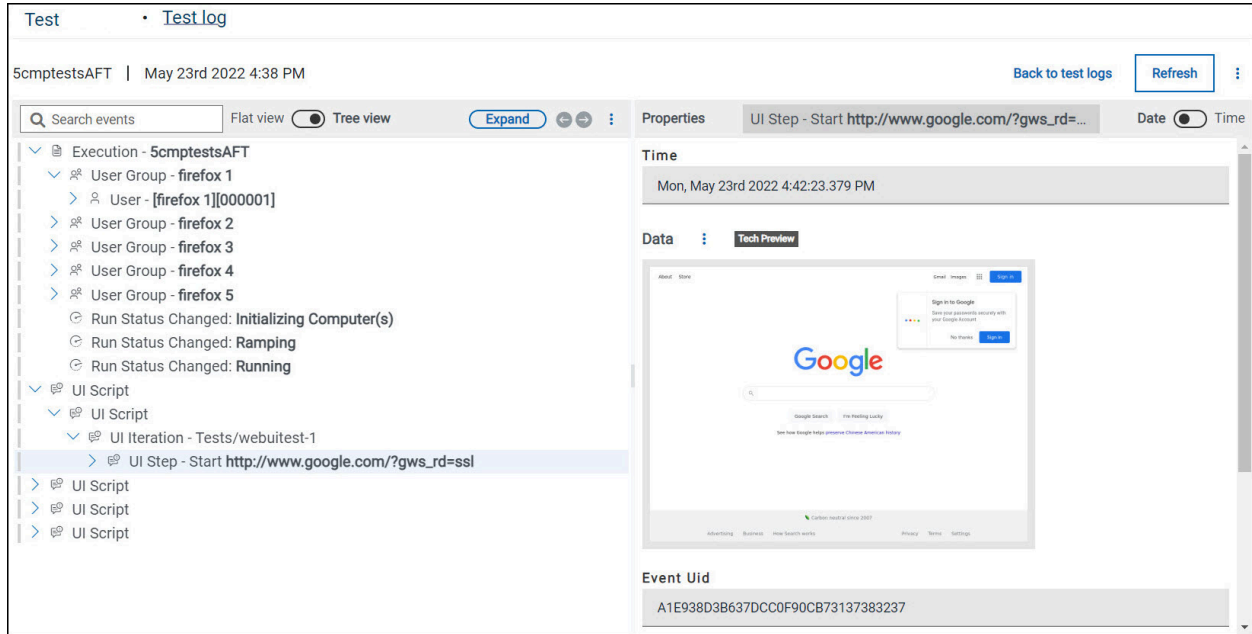
You can view the details of the test log and decide on the actions you might want to take on the test log. See [Test log details on page 815](#).

Test log details

You can find information about the details that are displayed for a test log on the **Test log** page. You can also find information to use the different options in the UI to change the view of the test log, find details of the events captured, navigate through the log, view or download attachments that are captured.

Before you can view the test log, you must have started a test run and opened the test log that is displayed on the **Test log** page.

An example of a test log that is displayed on the **Test log** page is as follows:



You can find the following details that are displayed on the **Test log** page:

Table 51. Test log details

Details about...	Examples as displayed in the Test log page...
The project name	An example of the project name that is displayed is as follows: <div style="text-align: center; font-size: 24px; font-weight: bold; color: #0070C0;">Test</div>
The name of the page	An example of the page name that is displayed is as follows: <ul style="list-style-type: none"> • Test log You can click the Test log link to view the test logs of all tests that ran in your project, if you are viewing the test log of a specific test run.
The name of the test asset	An example of the name of the asset that is displayed is as follows: <div style="text-align: center; font-size: 24px; font-weight: bold; color: #0070C0;">5cmptestsAFT</div>
The date and time when the test run was started from the Execution page	An example of the date and time when the test run was started that is displayed is as follows: <div style="text-align: center; font-size: 24px; font-weight: bold; color: #0070C0;">May 23rd 2022 4:38 PM</div>

Table 51. Test log details (continued)

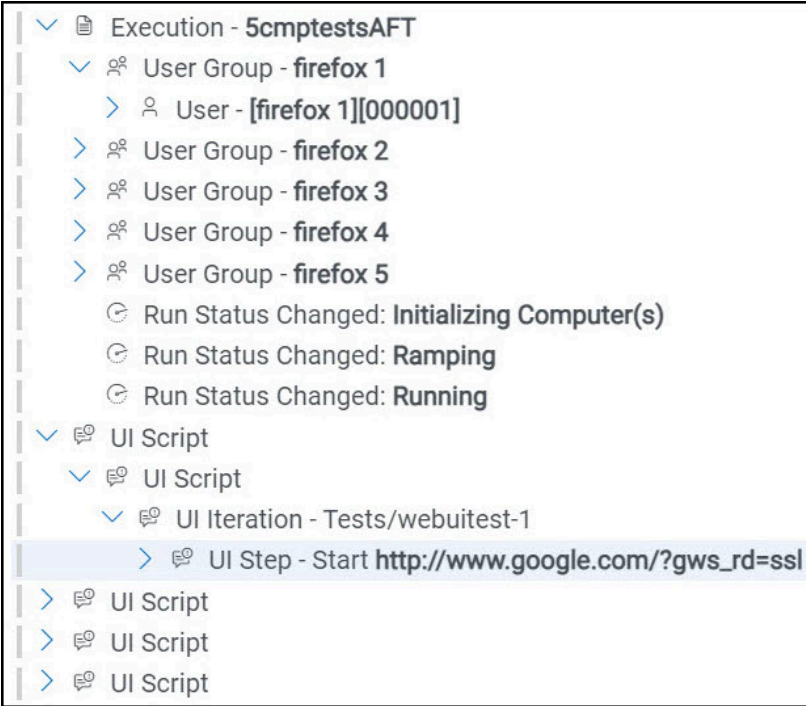
Details about...	Examples as displayed in the Test log page...
<p>The activities and events captured in the test log are displayed in the Tree view in the Test log panel.</p>	<p>An example of the log in the tree view that is displayed is as follows:</p>  <pre> Execution - 5cmptestsAFT ├── User Group - firefox 1 │ └── User - [firefox 1][000001] ├── User Group - firefox 2 ├── User Group - firefox 3 ├── User Group - firefox 4 ├── User Group - firefox 5 │ ├── Run Status Changed: Initializing Computer(s) │ ├── Run Status Changed: Ramping │ └── Run Status Changed: Running ├── UI Script │ └── UI Script │ └── UI Iteration - Tests/webuitest-1 │ └── UI Step - Start http://www.google.com/?gws_rd=ssl ├── UI Script ├── UI Script └── UI Script </pre>
<p>An event in the test log. Details of the event is displayed in the Properties panel.</p>	<p>An example of the details that is displayed in the Properties panel is as follows:</p>

Table 51. Test log details (continued)

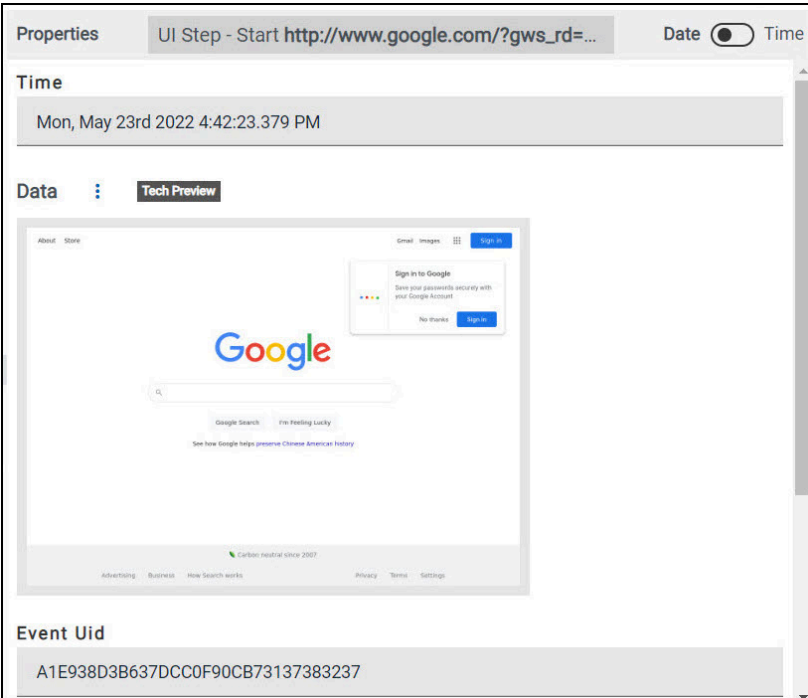



Details about...	Examples as displayed in the Test log page...
	
<p>The link to a URL that is recorded in an event. The URL is displayed as a link. When you click this link, the specific URL opens in a web browser.</p>	<p>An example of a link recorded in the event.</p> 
<p>An attachment of an event in the test log. The details of the attachment is displayed in the Properties panel.</p>	<p>An example of a text extracted from the attachment.</p> 
<p>The reason of a Verification Point (VP) that is captured in an event in</p>	<p>An example of the reason in a VP that is displayed in the Properties panel is as follows:</p>

Table 51. Test log details (continued)

Details about...	Examples as displayed in the Test log page...
the test log is displayed in the Properties panel.	

You can perform the actions detailed in the following table to change the view of the test log, find details of the events captured, navigate through the log, view or download attachments that are captured.

Table 52. Tasks that you can perform on the Test log page












Task	Action
<p>Changing the <i>Tree view</i> to a <i>Flat view</i> for the display of events in the test log panel.</p> <p> Note: The <i>Tree view</i> is the default view of the events in the test log.</p>	<p>Click  to toggle from the Tree view to the Flat view.</p> <p> Note: You can change the view from the one that is displayed by clicking .</p>
<p>Expanding the display of events in the test log panel.</p>	<p>Click  to expand the events that are contained within the event selected in the test log panel.</p> <p>For example, the collapsed view in the test log panel can be as follows:</p> <div data-bbox="667 1297 1341 1701" style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> >  Execution - 5cmptestsAFT ✓  UI Script <li style="background-color: #e6f2ff;">>  UI Script >  UI Script >  UI Script >  UI Script </div> <p>lows:</p>

Table 52. Tasks that you can perform on the Test log page (continued)

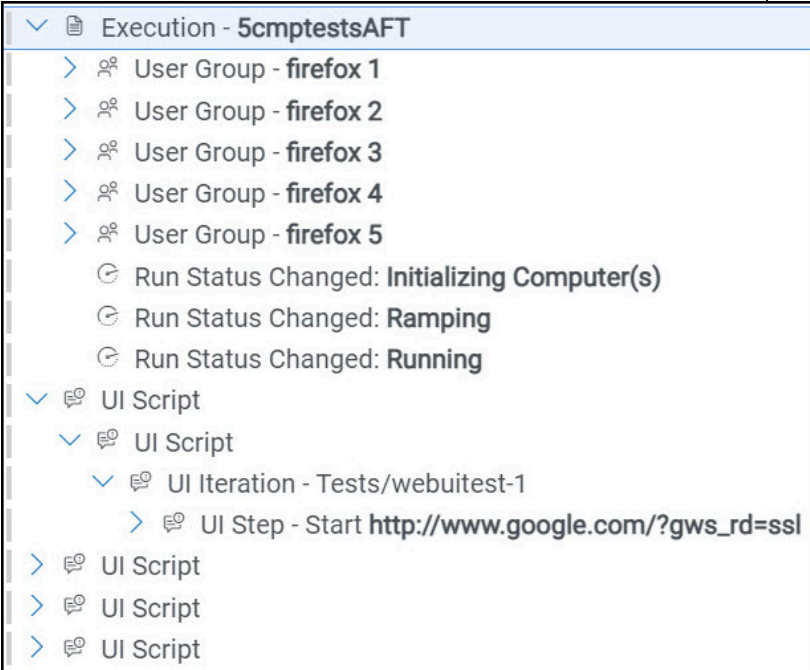



Task	Action
	<p>When the event is expanded, the view in the test log panel can be as follows:</p>  <p>low:</p>
<p>Collapsing the display of events in the test log panel.</p>	<p>Click  to collapse the events that are contained within the event selected in the test log panel.</p>
<p>Viewing primary activities when all activities are displayed.</p>	<p>Perform the following actions in the test log panel:</p> <ol style="list-style-type: none"> 1. Click the Open action menu  icon. 2. Click Show only primary activities. <p>The primary activities captured in the test log are displayed.</p>
<p>Viewing all activities when only primary activities are displayed.</p>	<p>Perform the following actions in the test log panel:</p> <ol style="list-style-type: none"> 1. Click the Open action menu  icon. 2. Click Show all activities. <p>All the activities captured in the test log are displayed.</p>
<p>Viewing failures captured in the test log.</p>	<p>The test log displays events marked with a colored vertical bar as follows:</p>

Table 52. Tasks that you can perform on the Test log page (continued)

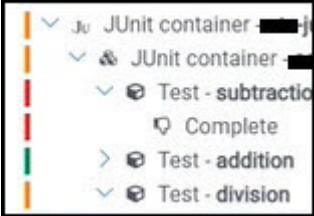


Task	Action
	<ul style="list-style-type: none"> • A green bar is marked for an event that passed in the test step. • An orange bar is marked for an event that had an error in the test step. • A red bar is marked for an event that failed in the test step. <p>For example, the view can be as follows:</p>  <p>You can navigate up and down in the view from one error or failed event to another one by using arrows  .</p>
Searching for events in the test log.	<p>In the search bar, you can enter text to find instances of events and properties that match with the text you entered.</p> <p>As a result of the search, the title of the nodes in the view and the names of the properties are highlighted. When the number of results found is displayed, you can press the F3 key to navigate from one search result to another in the tree view or flat view. You can press Ctrl and F3 to navigate from one result to another in the properties pane.</p>
Viewing the properties related to the event that you select in the test log panel.	<p>When you select an even in the test log panel, its properties are displayed in the properties panel.</p> <p>For example, the details that are displayed for an event can be as follows:</p>

Table 52. Tasks that you can perform on the Test log page (continued)

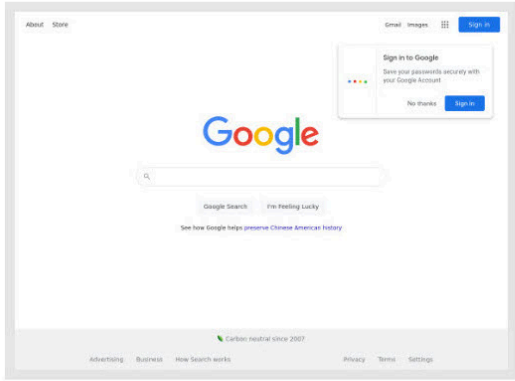
Task	Action
	<div data-bbox="613 310 1421 1003" style="border: 1px solid gray; padding: 5px;"> <p>Properties UI Step - Start http://www.google.com/?gws_rd=... Date <input type="checkbox"/> Time</p> <p>Time</p> <p>Mon, May 23rd 2022 4:42:23.379 PM</p> <p>Data Tech Preview</p>  <p>Event Uid</p> <p>A1E938D3B637DCC0F90CB73137383237</p> </div> <p>You can view the following information:</p> <ul style="list-style-type: none"> • The variable in the event is displayed. • The date stamp showing the time at which the event was captured in the test log. <p>You can toggle between the calendar time and the duration from the time the test started its run in the cluster by clicking the Date and Time toggle icon <input checked="" type="checkbox"/>. In the example, the date that is displayed in the Time field is as follows:</p> <div data-bbox="690 1432 1328 1587" style="border: 1px solid gray; padding: 10px; margin: 10px 0;"> <p>Time</p> <p>Mon, May 23rd 2022 4:42:23.379 PM</p> </div> <p>When the Time option is selected, the duration that is displayed in the Time field is as follows:</p> <div data-bbox="690 1709 976 1869" style="border: 1px solid gray; padding: 10px; margin: 10px 0;"> <p>Time</p> <p>00:03:42.177</p> </div>

Table 52. Tasks that you can perform on the Test log page (continued)

Task	Action
	<p>This time is the time that has elapsed from the start of the test run to the event in the test step.</p> <ul style="list-style-type: none"> • The name of the event. • The event ID. • Data that was captured as part of the test step. The items that are captured can be images, text files, PDF files, XML files, JSON files, and other supported formats. <p>You can view the attachment that is in the Data field, if it is supported to be viewed in a web browser by clicking the attachment. The attachment displays in a browser tab other than the one in which the Test log page is displayed.</p> <p>You can also copy the URL of the attachment or download a copy to your computer to use the attachment elsewhere.</p>
Viewing all test logs in the Test log page.	When you view the test log of a specific test asset and you want to return to the Test log page to view the test logs of the other test assets, you can click the Back to test logs option.

Test log queries

When a test run is complete in HCL DevOps Test Hub (Test Hub), if a test log is generated, you can run queries to extract and view test log data. The query is used to search for the events or activities in the test logs with a project or a result scope.

The query page contains a configuration section in which you enter and select search criteria.

The difference between an activity and an event are the following ones:

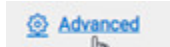
- An activity is characterized by a duration whereas an event is characterized by a particular time.
- An activity contains events.

When you define the search criteria and run a query, the returned test log data is displayed in a table. You can manage the data in the table by adding, moving, or deleting the columns.

Prerequisites

To get result log data queries, you must configure your tests with one of the following parameters to enable the test log and data query process:

- For JMeter, JUnit, Postman and HCL AppScan CodeSweep Suites, set the **history cisterna** parameter in the **Additional Configuration Parameters** field in the **Advanced** settings



- For API tests and Suites, AFT Suites and Functional test, and Compound Tests, set the **-history cisterna** parameter in the **Program Arguments** field in the **Advanced** settings.



Note: When you set this parameter, you can run your queries to collect test log data into Test Hub.

Query scope

A **Test result query** link is available in the test reports in the **Results** page when a test log is generated. If you click the link, it opens the query page in Test Hub where you can view all the events or all the activities for the current test result only. In the **Scope** field of the query page, you find the result id. You can run new queries with different properties as search criteria for this result.

You can also run new queries with different properties as search criteria for all the results in a project if you open the query page from **Analyze > Queries** in the navigation pane. In this case, the **Scope** field is at the project level.

In the **Queries** page, you can perform the following actions:

- Filter the events or activities.
- Sort your query results in an ascending or a descending order.
- Add, move, or delete fields in the columns of the table that displays the data of the query.
- Save a query, rename, reset or delete a query.
- Export a query.

You can export the queries to Shell, Curl, or PowerShell scripts. You can select the script, the output format, and the number of results that are returned by the query.

Configuring and running log queries

When a test run is complete and a test log created, you can configure and run queries to view test log data in HCL DevOps Test Hub (Test Hub) for one test log result or for all the test log results in a project.

Before you begin

You must have completed the following tasks:

- Configured and run a test. See [Test log queries on page 823](#).
- Opened the Queries page from the Results page or the navigation panel.

Scope: Result 2204

Query filters: Events Activities

Sort by: id ASC


Buttons: Copy Curl command, Add column, Run Query


id	time	type	properties
102150.2204	Dec 14, 2021 11:43 AM	Namespace::createResult	
102150.2204.A	Dec 14, 2021 11:44 AM	Element::createJMeterTest	

About this task


In the **Queries** page, you can configure a query and save it. You can customize the way the log data is displayed in the results table.

1. Click **Events** or **Activities**.
2. Enter properties and operators in the **Query filters** to filter the list of events or activities.

 **Note:** You can use the **Ctrl-Space** keyboard shortcut in this field to get the list of properties and operators that you can enter as filters to create your query.

 **Example:** Select **Activities** and enter 'type is JUnitTest' in **Query filters**.

3. Click **Run Query**.
4. Perform the following steps to save a query:
 - a. Click **Save as**.
 - b. Enter a name to identify the new query and click **Save**.

 **Example:** Enter 'My query'.

Result

The new query is displayed in the events and activities list.

5. Perform any of the following actions to configure the results table:


- Enter a property followed by the **ASC** operator or **DESC** in **Sort by** to sort the data in the table.



Note: **ASC** is used to sort the data from the lowest value to the highest value and **DESC** to filter the data in a descending order. You can use the **Ctrl-Space** keyboard shortcut to assist you. If you enter a combination of sort criteria, separate them with commas.



Example: Click the **Sort by** field, press **Ctrl-Space** and select in the list 'time' and enter 'ASC'.

- Click the action menu  in a column of the table to change the column display, and then perform any of the following actions:
 - Click **Insert new column**, enter a field name in **Search new column** and click **Add**.
 - Click **Move** to move a column before or after another one in the table.
 - Click **Remove** to remove a column from the table.
- Follow these steps to add a column to the end of the table:
 - a. Click **Add column**.





Result: A window opens.

- b. Enter a field name in **Search new column**.
- c. Click **Add**.

6. Click **Save** to save the configuration of the results table.

7. Follow these procedures to manage your query after it is created:

- Click the action menu  that is next to the **Save** button and select **Reset** if you have just modified a query and you want to undo your modification.
- Click the  action menu and select **Delete** to delete the new query from the **Events and activities** list.
- Click **Copy Curl command** if you want to export a query.



Note: The curl command is copied to the clipboard. You can paste it in a command interface to use the query in an external tool.

Machine learning capabilities in DevOps Test Hub

HCL DevOps Test Hub (Test Hub) uses Machine learning (ML) algorithms to analyze tests that match certain requirements and criteria, and then presents the findings as insights and recommendations after the tests are run. You can use the insights and recommendations to interpret the test results and identify the problem areas in the system under test.

Test Hub has incorporated ML capabilities to provide an automated means to identify problems in the system under test.

Tests analyzed through ML

The following types of tests are supported for analysis by using ML capabilities in Test Hub:

- Performance tests
- Schedules

Parameters analyzed through ML

The following parameters in Performance tests or Schedules are supported for analysis by using ML capabilities in Test Hub:


- The **Response Time Lock-Step Pattern** parameter.
- The **Response Time Standard Deviation Pattern** parameter.
- The **Throughput Drop Pattern** parameter.

Test Hub uses different analyzers for each of the parameters. For example, Test Hub uses the **Response Time Lock-Step Pattern** analyzer to analyze the tests for the **Response Time Lock-Step Pattern** parameter.

Criteria for analysis of the parameters

The following table lists the criteria used by the analyzers for analysis of the parameters in Test Hub:

Parameter analyzed	Criteria for analysis
Response Time Lock-Step Pattern	<p>The analyzer identifies the Response Time Lock-Step Pattern parameter in the overall page response time observed against the user count based on the following criteria:</p> <ul style="list-style-type: none"> • Tests have a minimum of 20 users. • Tests run for longer durations for a trend to be observed. • Tests contain a minimum of three different user groups and each group has an activity of more than two minutes.
Response Time Standard Deviation Pattern	<p>The analyzer attempts to detect the response time of pages that are more than thrice the value of the standard deviation calculated for the page response time as the Response Time Standard Deviation Pattern parameter.</p>
Throughput Drop Pattern	<p>The analyzer attempts to detect sudden drops in network throughput as the Throughput Drop Pattern parameter that is based on the following criteria:</p>

Parameter analyzed	Criteria for analysis
	<p data-bbox="521 275 1404 390"> Note: Sudden drops in throughput might be related to the performance tool itself, issues with network connectivity, or issues with signal-scalability of the system under test.</p> <ul data-bbox="573 453 1411 764" style="list-style-type: none"> • Tests have a minimum of 20 users. • Tests have stages with a reasonably high number of samples within a stage. • The length and intensity of the throughput drops that are less than 20% of the median observed within a stage are considered by the analyzer. When no throughput is received within 60 seconds, such drops are considered in the analysis. Short shark-tooth patterns in drops are not considered by the analyzer. • Parts of the stage with increasing users are not considered for analysis. <p data-bbox="586 800 1414 909">For example, when the test adds one user every minute and the user range is between one user to 500 users, the Throughput Drop Pattern parameter is not analyzed.</p>

Analysis at the project level

When you as a *Project Owner* or *Tester* configure a run of a Performance Test or a Schedule from your project in Test Hub, you must ensure that the criteria required for ML analysis are met. During the test run, Test Hub uses its ML capabilities to analyze the supported parameters and presents its findings as insights on the **Insights** page.

Any member of your project can view the insights by clicking **Analyze > Insights** from the navigation in the project or from the **Insights** section on the **Overview** page in the project.

Project members after viewing the insights can react by either agreeing or disagreeing with the insights suggested by Test Hub.

Analysers in a team space

You can view the list of analysers and description about the analyzers available in a Team space by clicking **Infrastructure > Analyzers** from the navigation in the team space. Also, you can enable or disable the analyzer, view the count of the reviewed insights and the review history of the insights generated by the analyzers.

After viewing the recommendation for a specific parameter, members of the project can alter the default threshold value that specifies the level of confidence required in the analysis of the parameter and run the tests again to improve the accuracy of the recommendations and insights.

You can also improve the accuracy of the recommendations by running multiple tests with differing loads repeatedly.

Using Machine learning capabilities

You can find information in the following documentation about using Machine learning (ML) capabilities in Test Hub:

- To understand the requirements of Performance tests or Schedules that are analyzed by using ML capabilities in Test Hub, see [Test run considerations for using Machine learning capabilities on page 379](#).
- To configure a run of a Performance Test, see [Configuring a run of a Performance test on page 605](#).
- To configure a run of a Schedule, see [Configuring a run of a Rate Schedule or VU Schedule on page 692](#).
- To view the ML insights into the Performance tests or Schedules in your project, which were analyzed after a run, see [Viewing Machine learning insights on page 829](#). You can also select your reaction to agree or disagree with the insights provided when you view the insight.
- To understand how to interpret the insights, see [Interpretation of insights on page 832](#).
- To view the ML analysis of the supported parameters when Performance tests or Schedules that are in different projects in your team space are run, see [Viewing Machine learning analyzers on page 834](#).
- To understand how to interpret the recommendations of the ML analysis of the supported parameters, see [Interpretation of parameter analysis on page 836](#).
- To view the reactions of project members to the ML insights, see [Viewing reactions marked by members to Machine learning insights on page 839](#).
- To modify the settings of the ML analyzer for a specific parameter, see [Modifying Machine learning analyzer settings on page 841](#).
- To delete the ML insights provided about the Performance tests or Schedules that were analyzed, see [Deleting Machine learning insights on page 843](#).
- To rerun the ML analysis without rerunning the test asset, see [Rerunning Machine learning \(ML\) analysis on page 844](#).

Viewing Machine learning insights

When you want to use Machine learning (ML) capabilities in HCL DevOps Test Hub (Test Hub), you need to run Performance Tests or Schedules that conform to certain criteria. After the test run is completed, you can view the ML insights on the **Insights** page in your project.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Ran the Performance Test or Schedule that matches the minimum requirements for ML analysis in Test Hub. See [Configuring a run of a Compound Test that contains performance tests on page 569](#) or [Configuring a run of a Rate Schedule or VU Schedule on page 692](#).

About this task

If the Performance Tests or Schedules match the requirements and criteria for ML analysis, Test Hub analyzes the test results for the relevant parameter and presents its findings as insights into the parameter.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Perform any of the following actions to view the insights:

- Scroll down the **Overview** page to see the **Insights** section.

Go to Step 4 on page 830 for the actions that you can perform to view the insights.

- Click **Analyze > Insights** in the navigation pane.

The **Insights** page is displayed.

Go to Step 5 on page 830 for the actions that you can perform to view the insights.

4. Perform the following steps to view a specific insight from the **Overview** page:

The **Insights** section on the **Overview** page displays a table of the insights for the supported parameters identified from the tests that were run in the project. The table lists the level of confidence with which the analysis of the specific parameter was done, the parameter analyzed, the name of the test asset, and the timestamp of the completed analysis.

- a. Click any item in the row of a specific insight to view the details of that insight.

The **Insights** page is displayed with the details of the specific insight.



Note: If you want to view the list of the other insights that are available in your project, you can click **Back to all insights**, and the list of available insights in the project is displayed in a tabular view.

- b. Go to Step 6 on page 831.

5. Perform the following steps to view a specific insight from the **Insights** page:

- a. Identify the test asset or the parameter by performing any of the following actions:

- Search for the test asset or parameter by entering any text contained in its name in the **Search** field.
- Sort the items in the table of insights by clicking any of the column headers, and then identify the insight.

For example, you can sort the items based on the name of the test asset in an ascending order or descending order, and then identify the test that you ran.

You identified the insight.

b. Click the **Expand** icon  of the insight.

The details of the specific insight are displayed.


c. Go to [Step 6 on page 831](#).


6. Perform the following steps when the details of the insight are displayed.


a. Read the details about the analysis and the issues that are identified by the analyzer.


b. Correlate the graph analyzed by the analyzer with your observations on the parameter.

c. Select your reaction to either agree or disagree with the insight by performing the following actions:

- Click the **Agree** icon , if you agree with the insights provided.

After you agree, the icon changes to  and displays the text *Agreed by username* when you hover over the icon.

- Click the **Disagree** icon , if you disagree with the insights provided.

After you disagree, the icon changes to  and displays the text *Disagreed by username* when you hover over the icon.



Note: A member of a project can only react once by either agreeing or disagreeing with the insight.

Results

You viewed the details of the insights into the parameters that are analyzed by Test Hub.

What to do next

You can perform any of the following tasks:

- Understand how to interpret the insights. See [Interpretation of insights on page 832](#).
- Delete an insight. See [Deleting Machine learning insights on page 843](#).
- Rerun the analysis so that the deleted instance of the insight is redisplayed. See [Rerunning Machine learning \(ML\) analysis on page 844](#).

Related reference

[Interpretation of insights on page 832](#)

Interpretation of insights

You can find information about interpreting the Machine learning (ML) insights into the supported parameters that are analyzed by HCL DevOps Test Hub (Test Hub) after certain tests are run.

Parameters analyzed through ML

The following parameters in Performance tests or Schedules are supported for analysis by using ML capabilities in Test Hub:

- The **Response Time Lock-Step Pattern** parameter.
- The **Response Time Standard Deviation Pattern** parameter.
- The **Throughput Drop Pattern** parameter.

Test Hub analyzes the parameters based on certain criteria as detailed in [Criteria for analysis of the parameters on page 827](#) and then presents its findings as insights about the parameters.

Details of the insights

The insights about the parameters analyzed are displayed in the **Insights** section on the project **Overview** page or on the **Insights** page in a project.

The insights analyzed for the parameters from various tests that were run in the project are listed in a table. The table lists the following information:

- Name of the test asset in the **Test** column.
- The parameter analyzed in the **Insight** column.
- The level of confidence with which the analysis of the parameter was done in the **Confidence** column. The level of confidence displayed for the analysis can be as *Lowest*, *Low*, *Medium*, *High*, or *Highest*. You can view the confidence level as a number when you hover the cursor over the text in the **Confidence** column.
- The timestamp of the completed analysis in the **Identified on** column.
- The labels that you added for the specific test run in the **Labels** column.
- The components in the System Model that are associated with the test asset in the **Components** column.

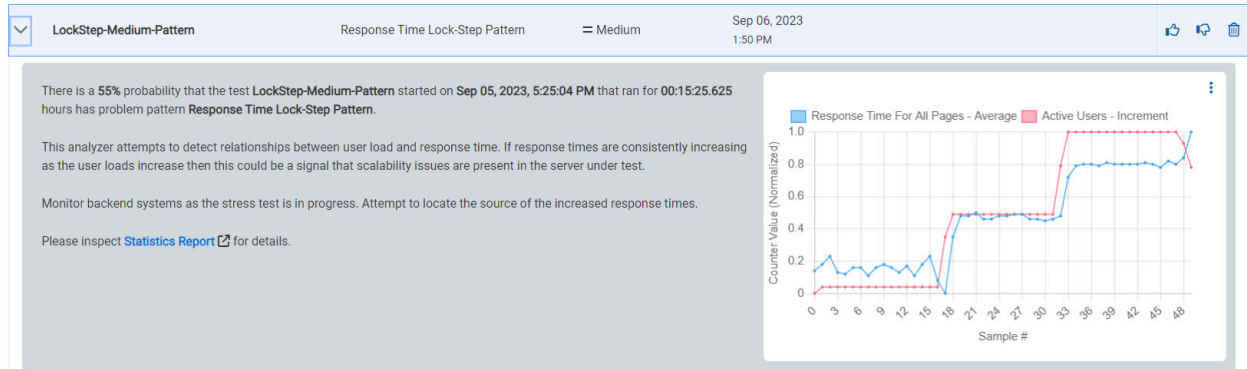
You can click the **Expand** icon  to view the details of the insight.

You can find information as an example about the following insights that help you to interpret the insight:

- [Insights on the Response Time Lock-Step Pattern parameter on page 832](#)
- [Insights on the Response Time Standard Deviation Pattern parameter on page 833](#)
- [Insights on the Throughput Drop Pattern parameter on page 834](#)

Insights on the Response Time Lock-Step Pattern parameter

When you expand the insight about the **Response Time Lock-Step Pattern** parameter, you can view the details displayed similar to the following sample:

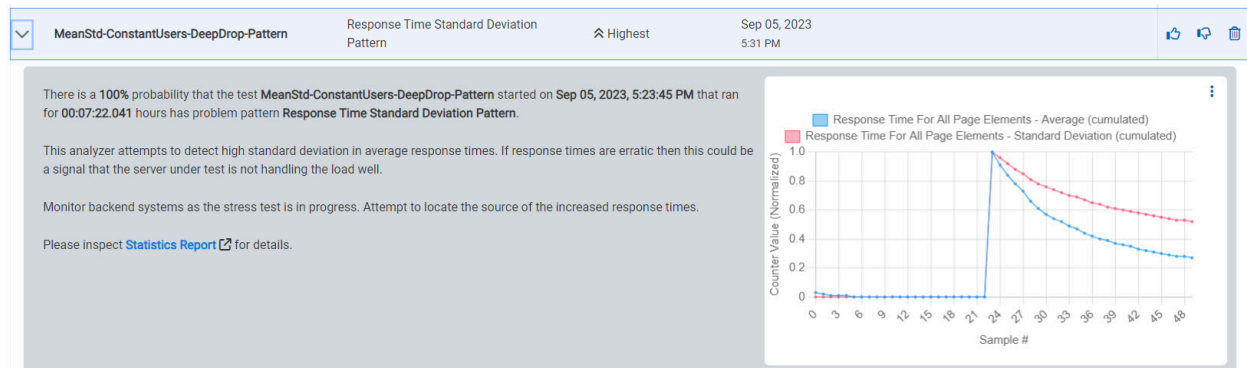


The following information is displayed as insights into the test on analyzing the **Response Time Lock-Step Pattern** parameter:

- The probability of finding a problem in the test.
- The description of how the ML analyzer has attempted to analyze the parameter in the test.
- The methods to control the problems are provided as suggestions.
- The graph of the observed data points.
- The link to the Statistics Report generated for the test.

Insights on the Response Time Standard Deviation Pattern parameter

When you expand the insight about the **Response Time Standard Deviation Pattern** parameter, you can view the details displayed similar to the following sample:

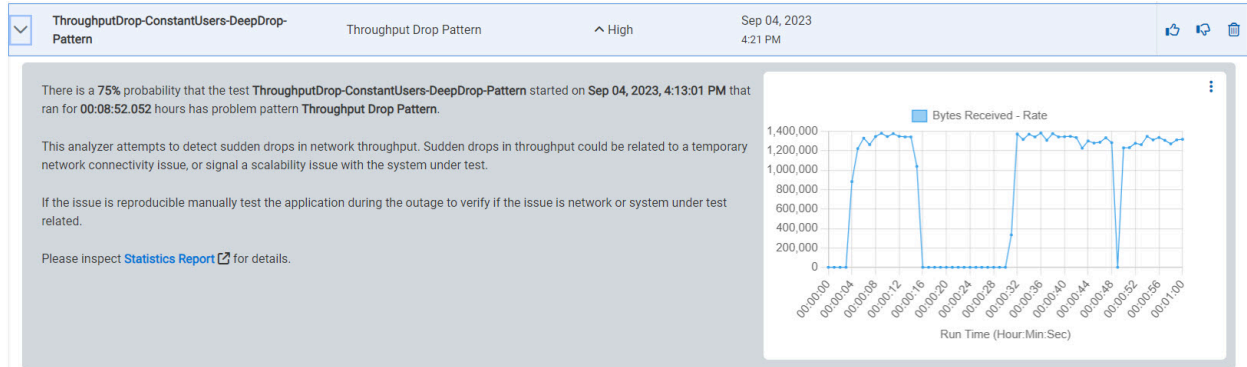


The following information is displayed as insights into the test on analyzing the **Response Time Standard Deviation Pattern** parameter:

- The probability of finding a problem in the test.
- The description of how the ML analyzer has attempted to analyze the parameter in the test.
- The methods to control the problems are provided as suggestions.
- The graph of the observed data points.
- The link to the Statistics Report generated for the test.

Insights on the Throughput Drop Pattern parameter

When you expand the insight about the **Throughput Drop Pattern** parameter, you can view the details displayed similar to the following sample:



The following information is displayed as insights into the test on analyzing the **Throughput Drop Pattern** parameter:

- The probability of finding a problem in the test.
- The description of how the ML analyzer has attempted to analyze the parameter in the test.
- The methods to control the problems are provided as suggestions.
- The graph of the observed data points.
- The link to the Statistics Report generated for the test.

Related information

[Viewing Machine learning insights on page 829](#)

Viewing Machine learning analyzers

HCL DevOps Test Hub (Test Hub) analyzes certain types of tests that are run in a team space by using its Machine learning (ML) capabilities. You can view the list of analysers and description about the analyzers available in a Team space.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Ran the Performance Test or Schedule that matches the minimum requirements for ML analysis in Test Hub. See [Configuring a run of a Compound Test that contains performance tests on page 569](#) or [Configuring a run of a Rate Schedule or VU Schedule on page 692](#).

About this task

If the Performance Tests or Schedules that are run from projects in a team space match the requirements and criteria for ML analysis, Test Hub analyzes the tests during their runs.










1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Infrastructure > Analyzers** in the navigation pane.

The **Analyzers** page is displayed.

You can view the parameters that are supported for ML analysis listed in a tabular format. The information displayed is as follows:

Column header	Description						
Status	Displays the status of the parameter analysis as active or inactive in the team space. The default value of the analysis is set as active or enabled.						
Name	Displays the name of the parameters that are supported for ML analysis, which is as follows: <ul style="list-style-type: none"> ◦ Response Time Lock-Step Pattern ◦ Response Time Standard Deviation Pattern ◦ Throughput Drop Pattern 						
Version	Displays the version of the ML analyzer.						
Last Run	Displays the date when the tests were run for the analysis.						
Unreviewed insights	Displays the number of insights into the specific parameter analyzed from different tests that are not reviewed by the project members.						
Actions	Displays the following icons for the actions that you can perform on each parameter analyzed: <table border="1" data-bbox="509 1356 1419 1824"> <thead> <tr> <th>Icon</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Displays the Edit Analyzer dialog when clicked. You can modify the analyzer settings for subsequent analysis.</td> </tr> <tr> <td></td> <td>Displays a dialog that contains the list of the reactions marked by different members on insights into the specific parameter when clicked. The agreement or disagreement reactions by members with the date and time of marking them are listed in chronological order in the dialog. You can view the insight on the Insights page in the project when you click .</td> </tr> </tbody> </table>	Icon	Description		Displays the Edit Analyzer dialog when clicked. You can modify the analyzer settings for subsequent analysis.		Displays a dialog that contains the list of the reactions marked by different members on insights into the specific parameter when clicked. The agreement or disagreement reactions by members with the date and time of marking them are listed in chronological order in the dialog. You can view the insight on the Insights page in the project when you click  .
Icon	Description						
	Displays the Edit Analyzer dialog when clicked. You can modify the analyzer settings for subsequent analysis.						
	Displays a dialog that contains the list of the reactions marked by different members on insights into the specific parameter when clicked. The agreement or disagreement reactions by members with the date and time of marking them are listed in chronological order in the dialog. You can view the insight on the Insights page in the project when you click  .						

3. Click the **Expand** icon  of the parameter to view the details.

The details of the parameter are displayed.

4. Read the details about the analysis and recommendations for troubleshooting or resolving the problems detected in the analysis.

See the information in the related links to understand how to interpret the analysis of the parameters.

Results

You viewed the details of the parameters that are analyzed by Test Hub in tests that were run in the team space.

What to do next

You can perform any of the following tasks:

- Understand how to interpret the recommendations from the analysis of the parameters. See [Interpretation of parameter analysis on page 836](#).
- View reactions by members to an insight. See [Viewing reactions marked by members to Machine learning insights on page 839](#).
- Modify the ML analyzer settings. See [Modifying Machine learning analyzer settings on page 841](#).

Related reference

[Interpretation of parameter analysis on page 836](#)

Interpretation of parameter analysis

You can find information about interpreting the details of each of the Machine Learning (ML) analyzers that analyze the parameters after certain tests are run in HCL DevOps Test Hub (Test Hub).

Parameters analyzed through ML

The following parameters in Performance tests or Schedules are supported for analysis by using ML capabilities in Test Hub:










- The **Response Time Lock-Step Pattern** parameter.
- The **Response Time Standard Deviation Pattern** parameter.
- The **Throughput Drop Pattern** parameter.


Test Hub analyzes the parameters based on certain criteria as detailed in [Criteria for analysis of the parameters on page 827](#). The ML analyzers analyze the results of tests run in all projects in a team space for each of the parameters. The analyzers present their findings on each of the parameters as an insight on the **Insights** page in the project.

Details of the parameters analyzed

The details about the parameters analyzed are displayed on the **Analyzers** page in the team space.

The parameters are listed in a table with the following information:

Column header	Description						
Status	Displays the status of the parameter analysis as active or inactive in the team space. The default value of the analysis is set as active or enabled.						
Name	Displays the name of the parameters that are supported for ML analysis, which is as follows: <ul style="list-style-type: none"> • Response Time Lock-Step Pattern • Response Time Standard Deviation Pattern • Throughput Drop Pattern 						
Version	Displays the version of the ML analyzer.						
Last Run	Displays the date when the tests were run for the analysis.						
Unreviewed insights	Displays the number of insights into the specific parameter analyzed from different tests that are not reviewed by the project members.						
Actions	Displays the following icons for the actions that you can perform on each parameter analyzed: <table border="1" data-bbox="446 1150 1421 1575"> <thead> <tr> <th>Icon</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Displays the Edit Analyzer dialog when clicked. You can modify the analyzer settings for subsequent analysis.</td> </tr> <tr> <td></td> <td>Displays a dialog that contains the list of the reactions marked by different members on insights into the specific parameter when clicked. The agreement or disagreement reactions by members with the date and time of marking them are listed in chronological order in the dialog. You can view the insight on the Insights page in the project when you click .</td> </tr> </tbody> </table>	Icon	Description		Displays the Edit Analyzer dialog when clicked. You can modify the analyzer settings for subsequent analysis.		Displays a dialog that contains the list of the reactions marked by different members on insights into the specific parameter when clicked. The agreement or disagreement reactions by members with the date and time of marking them are listed in chronological order in the dialog. You can view the insight on the Insights page in the project when you click  .
Icon	Description						
	Displays the Edit Analyzer dialog when clicked. You can modify the analyzer settings for subsequent analysis.						
	Displays a dialog that contains the list of the reactions marked by different members on insights into the specific parameter when clicked. The agreement or disagreement reactions by members with the date and time of marking them are listed in chronological order in the dialog. You can view the insight on the Insights page in the project when you click  .						



You can click the **Expand** icon  to view the details of the parameter analyzed.

You can find information as an example about the following parameters that help you to interpret the parameter analysis:

- [The Response Time Lock-Step Pattern parameter analyzer on page 838](#)
- [The Response Time Standard Deviation Pattern parameter analyzer on page 838](#)
- [The Throughput Drop Pattern parameter analyzer on page 839](#)

The Response Time Lock-Step Pattern parameter analyzer

When you expand the **Response Time Lock-Step Pattern** parameter, you can view the details displayed similar to the following sample:



Active	Response Time Lock-Step Pattern	0.9.1	September 6th 2023	3	 
Description:	This analyzer attempts to detect relationships between user load and response time. If response times are consistently increasing as the user loads increase then this could be a signal that scalability issues are present in the server under test.				
Recommendation:	Monitor backend systems as the stress test is in progress. Attempt to locate the source of the increased response times.				
Summary:	Detects correlations between user load and server response times.				
Counters:	Time [ms] Pages/Response Time/Average Response Time For All Pages For Interval/Mean Run/Active Users/!!REGEX!!Cumulated0*/Increment				
— Machine Learning Parameters					
Threshold: 0.275	Number of Epochs: 1	Batch Size: 1	Padding Size: 100		

The following information is displayed for the **Response Time Lock-Step Pattern** parameter:

- The description of how the ML analyzer has attempted to analyze the parameter in the test.
- The methods to control the problems are provided as a recommendation.
- The summary of the analysis.
- The counters used in the analysis.
- The parameters used in the analysis.

The Response Time Standard Deviation Pattern parameter analyzer

When you expand the **Response Time Standard Deviation Pattern** parameter, you can view the details displayed similar to the following sample:



Active	Response Time Standard Deviation Pattern	0.9.0	September 5th 2023	2	 
Description:	This analyzer attempts to detect high standard deviation in average response times. If response times are erratic then this could be a signal that the server under test is not handling the load well.				
Recommendation:	Monitor backend systems as the stress test is in progress. Attempt to locate the source of the increased response times.				
Summary:	Detects correlations between average page response time and the standard deviation of the page response time.				
Counters:	Pages/Response Time/Average Response Time For All Page Elements For Interval/Cumulated/Mean Pages/Response Time/Average Response Time For All Page Elements For Interval/Cumulated/StdDev Time [ms]				
— Machine Learning Parameters					
Threshold: 0.45	Number of Epochs: 1	Batch Size: 1	Padding Size: 1		

The following information is displayed for the **Response Time Standard Deviation Pattern** parameter:

- The description of how the ML analyzer has attempted to analyze the parameter in the test.
- The methods to control the problems are provided as a recommendation.
- The summary of the analysis.
- The counters used in the analysis.
- The parameters used in the analysis.

The Throughput Drop Pattern parameter analyzer

When you expand the **Throughput Drop Pattern** parameter, you can view the details displayed similar to the following sample:

Active	Throughput Drop Pattern	0.9.1	September 5th 2023	3	 
Description:	This analyzer attempts to detect sudden drops in network throughput. Sudden drops in throughput could be related to a temporary network connectivity issue, or signal a scalability issue with the system under test.				
Recommendation:	If the issue is reproducible manually test the application during the outage to verify if the issue is network or system under test related.				
Summary:	Detects unexplained sudden drops in network throughput.				
Counters:	Time [ms] Run/Bytes/Received/Rate Run/Active Users/!!REGEX!!Cumulated0*/Increment				
— Machine Learning Parameters					
Threshold: 0.5	Number of Epochs: 1	Batch Size: 1	Padding Size: 1		

The following information is displayed for the **Throughput Drop Pattern** parameter:

- The description of how the ML analyzer has attempted to analyze the parameter in the test.
- The methods to control the problems are provided as a recommendation.
- The summary of the analysis.
- The counters used in the analysis.
- The parameters used in the analysis.

Related information

[Viewing Machine learning analyzers on page 834](#)

Viewing reactions marked by members to Machine learning insights

You can view the reactions that members of a project mark to show their agreement or disagreement with the insights into the parameters on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Ran the Performance Test or Schedule that matches the minimum requirements for ML analysis in Test Hub. See [Configuring a run of a Compound Test that contains performance tests on page 569](#) or [Configuring a run of a Rate Schedule or VU Schedule on page 692](#).
- Viewed the ML insights and marked your reaction to an insight. See [Viewing Machine learning insights on page 829](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Analyze > Insights** in the navigation pane.

The **Insights** page is displayed.





4. Identify the test asset or the parameter by performing any of the following actions:



- Search for the test asset or parameter by entering any text contained in its name in the **Search** field.
- Sort the items in the table of insights by clicking any of the column headers, and then identifying the insight.

For example, you can sort the items based on the name of the test asset in an ascending order or descending order, and then identify the test that you ran.

You identified the insight.

5. Use the information in the following table to interpret the reactions marked for the insight:

If the icon is displayed as	Interpretation of the reaction
	None of the members agreed with the insight.
	At least one member agreed with the insight.  Note: The name of the member who reacted is displayed as <i>Agreed by username</i> when you hover over the icon.
	None of the members disagreed with the insight.

If the icon is displayed as	Interpretation of the reaction
	<p>At least one member disagreed with the insight.</p> <p> Note: The name of the member who reacted is displayed as <i>Disagreed by <code>username</code></i> when you hover over the icon.</p>

Results

You viewed the member reactions to an insight from the **Insights** page.

What to do next

You can perform any of the following tasks:

- Modify the threshold value of a specific parameter based on the number of disagreements marked by members so that you can improve the accuracy of the insight. See [Modifying Machine learning analyzer settings on page 841](#).
- Repeat the test runs with the updated thresholds to improve the Machine Learning insights for the specific parameter. See [Test run configurations on page 446](#).

Modifying Machine learning analyzer settings

When you want to change the decision threshold set for a Machine learning (ML) analyzer or disable the analyzer, you can modify the settings from the **Analyzer** page.

Before you begin

You must have completed the following tasks:


- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Ran the Performance Test or Schedule that matches the minimum requirements for ML analysis in Test Hub. See [Configuring a run of a Compound Test that contains performance tests on page 569](#) or [Configuring a run of a Rate Schedule or VU Schedule on page 692](#).
- Viewed the insights into tests provided by ML analysis. See [Viewing Machine learning insights on page 829](#).
- Viewed the reactions from members of your project to the ML insight. See [Viewing reactions marked by members to Machine learning insights on page 839](#).
- Viewed the history of member reactions for a specific parameter analyzer from the **Analyzers** page. See [Viewing Machine learning analyzers on page 834](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.



2. Click **Infrastructure > Analyzers** in the navigation pane.

The **Analyzers** page is displayed. You can view the parameters that are supported for ML analysis listed in a tabular format.

- Identify the parameter whose analyzer you want to modify, and then click the **Edit Analyzer** icon  in the **Actions** column in the row of the parameter.

The **Edit Analyzer** dialog is displayed.

- Perform any of the following actions to modify the default settings for the ML analysis of the specific parameter.

If you want to...	Action
Disable the analysis for the specific parameter.	Perform the following steps: <ol style="list-style-type: none"> Click Is Enabled to switch the analysis from being enabled or active to become disabled or inactive. <p> Note: If you modify the default setting to be disabled, Test Hub does not perform ML analysis of the specific parameter when tests are run subsequently.</p> <ol style="list-style-type: none"> Click Submit.
Enable the analysis for the specific parameter, if it is disabled.	Perform the following steps: <ol style="list-style-type: none"> Click Is Enabled to switch the analysis from being disabled or inactive to become enabled or active. Click Submit.
Increase or decrease the threshold level.	Perform the following steps: <ol style="list-style-type: none"> Edit the value in the Decision Threshold field and enter a value as required. <p>The threshold value determines the level after which the ML analysis identifies an insight into the specific parameter when the test runs.</p> <p> Note: You can retain the default values for the decision threshold initially until a large number of tests are run that contribute to the insights for a specific parameter. You can change the value to a higher threshold when there are a large number of records learned by the ML analyzers.</p> <ol style="list-style-type: none"> Click Submit.

Results

You either disabled or enabled the analyzer or changed the decision threshold set for a ML analyzer.

What to do next

You can perform any of the following tasks:

- Rerun the analysis of the test with the changed threshold values without running the tests again. See [Rerunning Machine learning \(ML\) analysis on page 844](#).
- Run tests if you changed the decision threshold values so that the ML analysis uses the changed values. See [Test run configurations on page 446](#).
- View insights into the specific parameter whose decision threshold was modified after the test runs are completed. See [Viewing Machine learning insights on page 829](#).
- View the ML recommendations for the specific parameter whose decision threshold was modified. See [Viewing Machine learning analyzers on page 834](#).

Deleting Machine learning insights

After you view the Machine learning (ML) insights into the tests, you might want to delete the insight if many members of your project disagree with the insight.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.
- Viewed the insights into tests provided by ML analysis. See [Viewing Machine learning insights on page 829](#).
- Viewed the reactions from members of your project to the ML insight. See [Viewing reactions marked by members to Machine learning insights on page 839](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.


The **Overview** page of the project is displayed.

3. Perform any of the following actions to view and identify the insight:

- Scroll down the **Overview** page to see the **Insights** section.

The **Insights** section on the **Overview** page displays a table of the insights for the supported parameters identified from the tests that were run in the project. You can identify the insight and click any item in the row of the identified insight to display the insight on the **Insights** page.

- Click **Analyze > Insights** in the navigation pane. The **Insights** page is displayed. You can identify the insight that you want to delete.

4. Click the **Delete** icon  from the **Actions** column that is in the row of the identified insight.

The **Delete Insight** dialog is displayed.

5. Click **Yes**.

The insight is deleted and a confirmation message is displayed.

Results

You deleted an insight into the test arrived at by ML analysis of the test.

What to do next

You can perform any of the following tasks:

- Run tests to let the ML analyzers reanalyze and present the insights. See [Test run configurations on page 446](#).
- Rerun the analysis without running the tests again so that the deleted instance of the insight is redisplayed. See [Rerunning Machine learning \(ML\) analysis on page 844](#).

Rerunning Machine learning (ML) analysis

If you change the decision threshold value set for a parameter analyzer, you can then rerun the analyzer to analyze the parameter for the changed threshold and present the insight without running the tests again.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Analyze > Results** from the navigation pane in your project.

The **Results** page is displayed.

4. Click the **SUITES** tab, if you ran a Schedule or the **TESTS** tab, if you ran a Performance Test.

5. Identify the test result for which you want to rerun the analysis by performing any of the following actions:

- Search for the test result by entering any text contained in the test asset name in the **Search** field.
- Sort the items in the table of results by clicking the **Started by** column header, and then identifying the test runs that you started.

6. Click the **Open action menu** icon  from the **Actions** column that is in the row of the identified test result.

7. Click **Rerun analysis** .

The parameter is analyzed again with the changed threshold value and the insight presented.

Results

You started the ML analysis of the previously analyzed test result so that the insight can be displayed based on the changed threshold value without running the test again.

Resource monitoring

When you apply load to a system under test, the amount of resources consumed by your system is increasing. If the capacity of the resources does not match the load, you can see performance issues in the results. Resource monitoring in HCL DevOps Test Hub (Test Hub) helps you monitoring the resources of a system and establish the performance metrics and usage data of the system. Thus, you can observe the health of these resources while a schedule is running.

You can use resource monitoring while running a schedule to capture data, such as processor or memory usage, or to monitor the availability of hosts and services by using counters. Test Hub provides a comprehensive view of a system under test to help determine problems. Hosts and services can be servers, virtual machines, or any local host or network services. You can also monitor remote hosts and services with agents.

Monitoring a host or a server

You can use resource monitoring to monitor any host and network service. This method is used for host targets that the service can reach directly. With Test Hub, you can also monitor performance metrics that are collected and stored by a monitoring system that is monitoring a host. You can use a Prometheus server or a data collector through an exporter, OpenMetrics exporter, for example.

To monitor resources on a local system, you must add the source of the resource monitoring data from the resource monitoring page. Then you have to enter some connection settings and select the counters to monitor the sources. The metrics and usage data are exposed in a graph for each selected counter.

Monitoring a remote host with monitoring agents

You can monitor remote hosts and services with agents through wider sets of data collectors, computers, and networks. The purpose is to capture CPU load, disk space, memory, and the running process for example.

Agent-based monitoring is useful when remote services are not directly accessible from Test Hub through the network. Agents are closer to the target that you want to monitor. You can set up the agent on an authorized host, for example, when access to an Apache httpd or NGINX server status page or a JVM JMX port, a Prometheus server, or an OpenMetrics exporter is restricted to one or few client hosts only. The configuration task is simplified, and no security changes are required.

Agent-based monitoring is mandatory to collect data from a Windows or Linux Performance host or from a Docker host.

To monitor a host across a monitoring agent, you must first set up a resource monitoring agent on the target host for which you want to collect the performance statistics and usage data. The agent establishes a connection with Test Hub. When the connection is set, the agent is showing up in the list of agents on the resource monitoring main page. Then, you can select the agent in the list of target sources in the dialog **Access target from** in the dialog that opens from **Add a Source**.

To learn more about configuring the resource monitoring sources, counters, and agents and collecting and viewing the data in Test Hub, go through the information in the following links.

Resource monitoring capabilities

With HCL DevOps Test Hub (Test Hub), you can monitor a host or a service that you add to a project as a source. You can select the statistics counters that are used to collect metrics and the resource usage data throughout the monitoring system. You can visualize metrics and data that are gathered during the monitoring process in Test Hub. You can add the resource monitoring sources and counters in a team space or in a project to collect metrics and resource usage data.

Requirements:

- You must be a *Team Space Owner* to create, modify and remove resource monitoring sources in a team space. Any team space *Member* (not owner) can only view resource monitoring sources.
- You must be a project *Owner* to create, modify and remove resource monitoring sources in a project. Any project *Member* (not the owner) can only view resource monitoring sources.

Supported sources

With Test Hub, you can monitor resources for the following sources:

- **Apache httpd server:** You can optionally use agents.
- **NGINX and NGINX Plus:** You can optionally use agents.
- **Java Virtual Machine:** You can optionally use agents.

You can monitor JVM resources from a local or a remote system. Some parameters must be set in the command before running the Java Virtual Machine. For more details, see the link to the page about starting a Java Virtual Machine on this page.

- **Windows Performance host:**

To monitor the performance of a Windows host, you must have installed an agent on the target Windows host

or on a Windows host that is configured to access the performance data of the Windows host target.

- **Linux Performance host:**

To monitor the performance of a Linux host, you must have installed an agent on the target Linux host.

- **Docker host:**

A resource monitoring agent is mandatory to get a Docker data collector. The Docker host source is added to the list of resource monitoring data collectors when the agent is installed.

- **Prometheus server:**

You can monitor metrics of a host under test that are collected by a Prometheus server. Prometheus collects metrics from monitored targets by regularly requesting appropriate HTTP endpoints on these targets (called *scraping*).

The metrics data are collected through data collectors sets (groups of monitoring counters) that are tracking the system performance. The performance data results are stored in the Prometheus database so that they can be consumed by external systems through a REST API.

In Test Hub, you can query these performance metrics collected by Prometheus servers. 'PromQL' is the language for querying Prometheus metric data. Refer to Prometheus PromQL documentation for information about this query language.

You can select default queries, or create additional ones.

A Prometheus server is required. It must be configured to scrap a set of exporters, every 15 seconds by default. Pushgateway, service discovery, and Alertmanager are optional.

If Test Hub cannot directly reach the Prometheus server, you can consider setting up an agent on an authorized host or on the same host that is running Prometheus.

- **OpenMetrics exporter:**

You can monitor the metrics of a host under test that are collected and exposed through an OpenMetrics format or a Prometheus exporter format.

Test Hub can scrap metrics exposed by an OpenMetrics or Prometheus exporter through metric counters. No Prometheus server installation is required, you only need to set up one or multiple OpenMetrics, or Prometheus exporters that fit your software or host target.

If Test Hub cannot directly reach the Prometheus server, you can consider setting up an agent on an authorized host or on the same host that is running Prometheus.

Monitoring an NGINX server

To monitor an NGINX server, you must add the NGINX server source to HCL DevOps Test Hub (Test Hub), enter connection settings and select the counters that are used to capture the monitoring usage data.

Before you begin

You must have logged in Test Hub and be the owner of an existing new project.

If you monitor a remote host, it must be connected with the computer that you use to access resource monitoring. See [Resource monitoring agents on page 864](#).

You can optionally use agents.

About this task

This task applies to NGINX and NGINX Plus.

If you use an agent to connect

1. Click **Infrastructure > Resource Monitoring**.

The resource monitoring page opens.

2. In the resource monitoring page, click **Add a Source** and select **Add an NGINX server...**

3. Fill in the following connection settings in the **New server** dialog:

- a. In **Target host**, enter the IP address or hostname and port number of the host where the server to monitor is installed.



Note: For hosts that are already connected to Test Hub through an agent, you have only the **Access target from** field enabled. Move directly to step 4.

- b. In **Server edition**, select the appropriate server from the drop-down list.

- c. In **Path to the status page of the server**, enter the name of the page to view the status of the server.



Note: If you select NGINX Plus (with API version 3) as a source, you must specify the name of the path to view the API root of the NGINX Plus server.

- d. In **Security**, select the following options:

Secured with TLS/SSL

If the application server is secured with TLS/SSL.

Trust self-signed certificate

To accept the server certificate.

Do not verify hostname

To ignore verification of the hostname in the certificate.

Require credentials

If the server requires log-in credentials, enter a **User name** and **Password**.

- e. Click **Add**.

The **Choose Data Counters** dialog opens.

4. Select and save the statistics counters to monitor the source. You can select them from the list where they follow the server logical organization.

For a faster selection, select the counters from the built-in sets drop-down list where they are organized by theme and save your selection.

HTTP Server Zones (8/8)
HTTP Server Zones Errors (2/2)
▼

Connections and Request

HTTP Server Zones

HTTP Server Zones Errors

Stream Server Zones

Stream Server Zones Errors

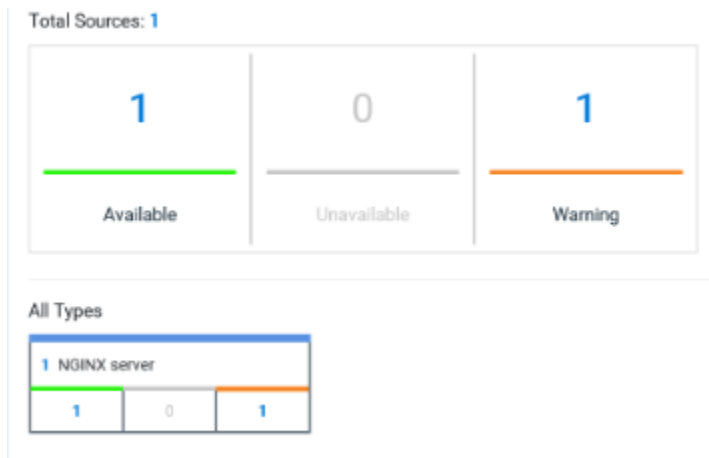
**Note:**

For Docker host, you can select different sets of counters.

- The first level of counters: Generic counters that are related to a Docker Image and all its running Docker Containers (in an exit, running or paused, created, restarted, removing, dead or transitive state).
- The second level of counters: Specific counters that are related to the existing Docker Containers.

Results

When the selected counters are saved, two tables are displayed in the resource monitoring main page. They contain the total number of sources you have added and the number of sources ordered by type.

**What to do next**

You can click the links in the tables to view the performance metrics of your monitored system.

Related information

[Viewing resource monitoring usage data on page 885](#)

Monitoring an Apache httpd server

To monitor an Apache httpd server, you must add the Apache httpd server source to HCL DevOps Test Hub (Test Hub), enter connection settings and select the counters that are used to capture the monitoring usage data.

Before you begin

You must be logged in Test Hub and be the owner of an existing new project.

You can optionally use agents. See [Resource monitoring agents on page 864](#).

1. Click **Infrastructure > Resource Monitoring**.

The resource monitoring page opens. See [Resource monitoring agents on page 864](#).

2. In the resource monitoring page, click **Add a Source** and select **Add an Apache httpd server...**
3. Fill in the following connection settings in the **New server** dialog:



Note: For hosts that are already connected to Test Hub through an agent, you have only the **Access target from** field enabled. Move directly to step 4.

- a. In **Target host**, enter the IP address or hostname and port number of the host where the server to monitor is installed.
- b. In **Server edition**, select the appropriate server from the drop-down list.
- c. In **Path to the status page of the server**, enter the name of the page to view the status of the server.
- d. In **Security**, select the following options:

Secured with TLS/SSL

If the application server is secured with TLS/SSL.

Trust self-signed certificate

To accept the server certificate.

Do not verify hostname

To ignore verification of the hostname in the certificate.

Require credentials

If the server requires log-in credentials, enter a **User name** and **Password**.

4. Click **Add**.

The **Choose Data Counters** dialog opens.

5. Select and save the statistics counters to monitor the source. You can select them from the list where they follow the server logical organization.

For a faster selection, select the counters from the built-in sets drop-down list where they are organized by theme and save your selection.

Results

When the selected counters are saved, two tables are displayed in the resource monitoring main page. They contain the total number of sources you have added and the number of sources ordered by type.

What to do next

You can click the links in the tables to view the performance metrics and resource monitoring usage data for the monitored server.

Related information

[Resource monitoring on page 845](#)

[Viewing resource monitoring usage data on page 885](#)

Monitoring metrics collected by a Prometheus server

You can monitor your system resources by using metrics data that are collected by a Prometheus server while executing a schedule.

Before you begin

You must have created a project in HCL DevOps Test Hub (Test Hub).

You must have installed and configured a Prometheus server.

Read the concept information about resource monitoring and monitoring Prometheus server source. See [Resource monitoring on page 845](#)

About this task

The following procedure describes how to select and create Prometheus queries in Test Hub. You must set a connection between Test Hub and the Prometheus server first to access to the Prometheus queries.

Enter the connection settings, then select queries.

1. Click **Add a source** and select **Add a Prometheus Server**.
2. Enter the connection settings in the **New server** dialog box.
 - a. In **Target host**, enter the IP address or host name and port number of the host where the Prometheus server is installed.
 - b. Enter the **Path to the API root of Prometheus** to point to the Prometheus API where the monitoring data are saved.

- c. Enter the security parameters if required. For information about this section, see related links.
- d. Click **Add** to close the dialog.

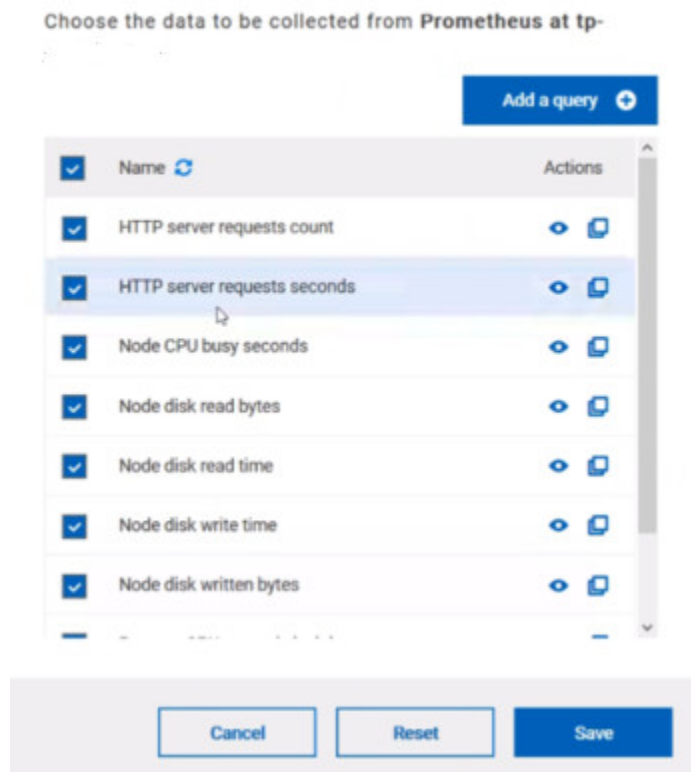
Result

The Prometheus server source is created successfully in the background and a dialog opens to select the Prometheus data queries.

3. In the **Select Prometheus queries** dialog, select queries that are available in the list to collect Prometheus metrics data by clicking the checkboxes.

You can reset the selection.

Select Prometheus queries



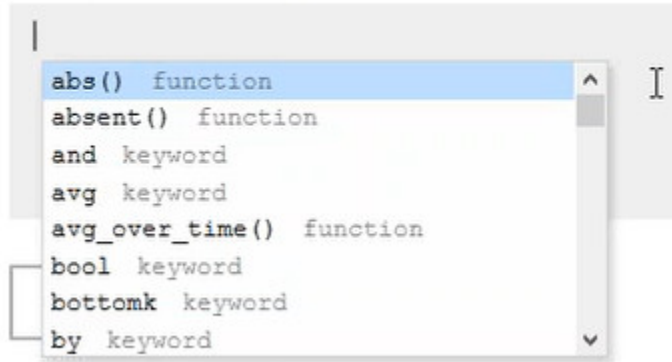
4. Follow these steps to create a query:

- a. Click **Add a query**.
- b. Enter a name for the new query in the **Create a Prometheus query** dialog.
- c. Enter a PromQL query.any keyword and press Ctrl+Space in the **Type and test your PromQL query**.

Result

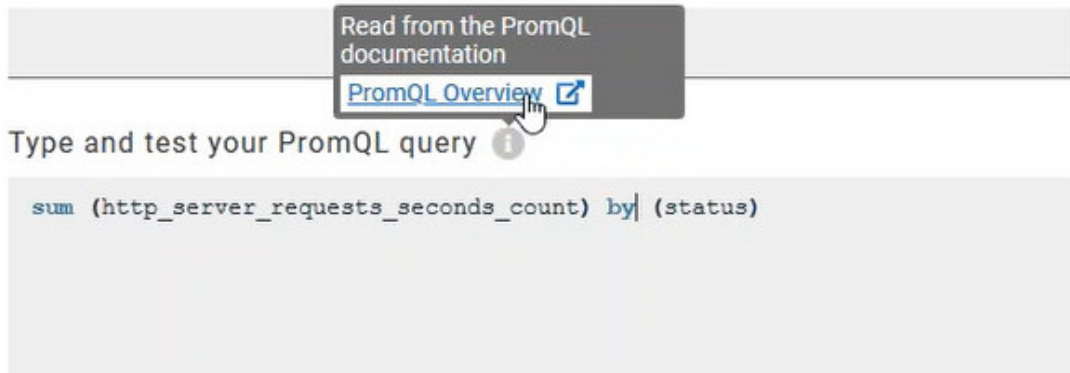
All the possible combinations of keywords and metrics related to your searched keyword are listed.


Type and test your PromQL query ⓘ



- d. Select a query in the list.
- e. Click the link that points to the 'Querying Prometheus documentation' page in the information tool tip to find other examples of queries in a web page that you can use as samples.

Name



- f. Click **Test query**.
If the test is successful, a green check is displayed, otherwise you get a red check with an error message.
 - g. Click **Create**.
5. Proceed as follows to duplicate a query:
 - a. Click the **Duplicate** icon  to duplicate a standard query.
 - b. Rename the duplicated query in the **View the Prometheus query** dialog.
Make sure that the name you give to the query is not already used.
 - c. Modify the query type.

- d. Test the query .
- e. Click **Create**.



Note: You can click Create or Duplicate only if the test of the query is successful and the name of the query is not already in use.

Create a Prometheus query


Name

JVM Memory

Type and test your PromQL query ?

```
jvm_memory_committed_bytes
```

✔

6. Proceed as follows to modify a query:
 - a. Click the **Modify** icon  .
 - b. Modify the name and type of the query.
 - c. Test the query.
 - d. Click **Update**.
 - e. Delete duplicated or created queries.
 - f. Click **Save** to close the **Select Prometheus queries** dialog.
7. Click Save and close the **Select Prometheus queries** dialog.

Results

The resource monitoring page displays the total number of sources you have added to your project and the number of sources ordered by type in cards.

What to do next

Click the links in the cards to go to the sources page and see the performance metrics for each source you added to your project. For more details, see [Viewing resource monitoring usage data on page 885](#)

Related information

[Resource monitoring on page 845](#)

[Viewing resource monitoring usage data on page 885](#)

Monitoring metrics exposed by an OpenMetrics exporter

With HCL DevOps Test Hub (Test Hub), you can monitor your system resources by using metrics data that are exposed by an OpenMetrics or a Prometheus exporter through metric counters while executing a schedule.

Before you begin

You must have at least one exporter or software to expose monitoring metrics with an OpenMetrics or Prometheus format.

You must have created a project in Test Hub.

No Prometheus server installation is required, you only need to set up one or multiple OpenMetrics, or Prometheus exporters that fit your software or host target. Refer to the link at the end of this page to see the list of exporters.

About this task

The following procedure describes how to select and create OpenMetrics counters in Test Hub. You must set a connection between Test Hub and the OpenMetrics exporter to select counters.

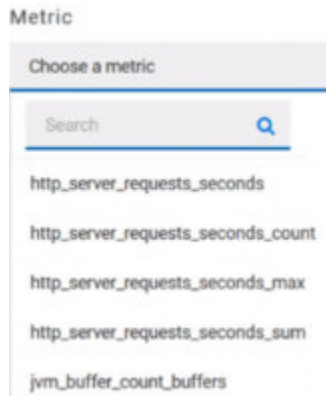
1. Click **Add a source** and select **Add an OpenMetrics exporter**.
2. In the **New OpenMetrics exporter** dialog that opens, enter the OpenMetrics exporter details to access its exposition endpoint as follows:
 - a. In **Target host**, enter the IP address or host name and port number of the host where the server is installed.
 - b. Enter the **Path to the exposition endpoint** to point to the application where the data performance results must be stored. As the path depends on the type of exporter used, refer to the exporter's documentation and configuration to indicate the right path.
 - c. In **Security**, enter the security parameters if required.
 - d. Click **Add** to close the dialog.

Result

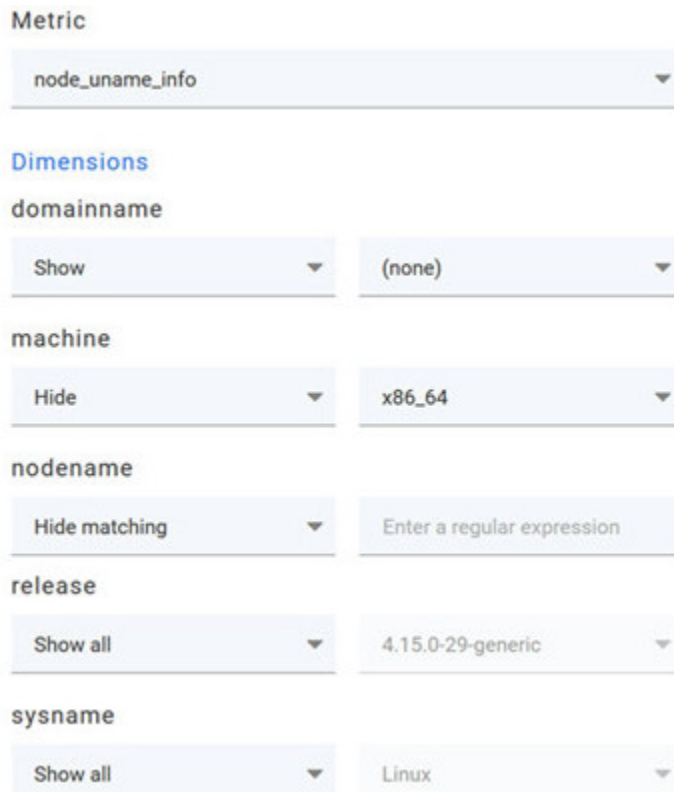
The source is created successfully in the background. A dialog box opens to select the OpenMetrics counters.

3. Click the checkboxes to select counters in the **Select OpenMetrics counters** dialog.
4. Follow these steps to create a counter:

- a. Click **Add a counter**.
- b. Enter a name for the new counter, in the **OpenMetrics counter**.
- c. Select metrics in the drop-down list or enter a keyword in the **Dimensions** field to enable the dynamic input help. You can also use the search field that appears in the list of metrics.



- d. Select **Dimensions** from the dynamic fields when they are required.



- e. Select the appropriate OpenMetrics counter that you are looking for.
- f. Click **Create**.

Create an OpenMetrics counter

Name

JVM Memory

Metric

jvm_memory_used_bytes

Dimensions


area


Show all heap,nonheap

id


Show all 6 items selected

Cancel Create

5. Follow these steps to duplicate a counter:
 - a. Select a counter in the list, and click the **Duplicate** icon .
 - b. Modify the name of the counter and the Dimensions.
 - c. Click **Create**.
6. Proceed as follows to modify a new counter:
 - a. Click the **Modify** icon.
 - b. Modify the name, Metrics, and Dimensions of the counter in the **Modify the OpenMetrics counter**.
 - c. Click **Update**.

 **Note:** You can create, duplicate, or update a counter only if the name of the counter is not already used.

7. Click **Save** and close the **Select OpenMetrics counters** dialog.

 **Note:** You can delete counters, but only the ones that you created or duplicated.

Results

The resource monitoring page displays the total number of sources you have added to your project and the number of sources ordered by type in cards.

What to do next

Click the links in the cards to go to the sources page and see the performance metrics for each source you added to your project. For more details, see [Viewing resource monitoring usage data on page 885](#).

Related information

[Exporters and Integrations](#)

[Resource monitoring on page 845](#)

Monitoring a Java Virtual Machine

To monitor a Java Virtual Machine, you must add the Java Virtual Machine source to the project in HCL DevOps Test Hub (Test Hub), and then select the counters that are used to capture the monitoring metrics.

Before you begin

You must have completed the following tasks:

- Configured and started the Java Virtual Machine from a command to connect it Test Hub.
See [Starting a Java Virtual Machine on page 875](#).
- Logged in Test Hub. You must be the owner of an existing new project.
- You can optionally use agents. See [Resource monitoring agents on page 864](#).

About this task

The Java Virtual Machine source is added to the list of resource monitoring sources when the agent is installed.

1. Click **Infrastructure > Resource Monitoring**.
The resource monitoring page opens.
2. In the resource monitoring page, click **Add a Source** and select **add a Java Virtual Machine**.
3. Fill in the following connection settings in the **New server** dialog:



Note: For hosts that are already connected to Test Hub through an agent, you have only the **Access target from** field enabled. Move directly to step 4.

- a. In **Target host**, enter the IP address or hostname and port number of the host where the server to monitor is installed.
- b. In **Server edition**, select the appropriate server from the drop-down list.
- c. In **Path to the status page of the server**, enter the name of the page to view the status of the server.

d. In **Security**, select the following options:

Secured with TLS/SSL

If the application server is secured with TLS/SSL.

Trust self-signed certificate

To accept the server certificate.

Do not verify hostname

To ignore verification of the hostname in the certificate.

Require credentials

If the server requires log-in credentials, enter a **User name** and **Password**.

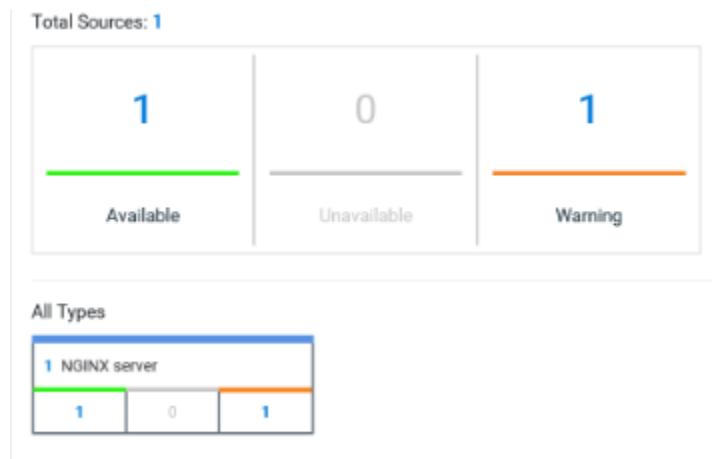
4. Click **Add**.

5. Select and save the statistics counters to monitor the source. You can select them from the list where they follow the server logical organization.

For a faster selection, select the counters from the built-in sets drop-down list where they are organized by theme and save your selection.

Result

When the selected counters are saved, two tables are displayed in the resource monitoring main page. They contain the total number of sources you have added and the number of sources ordered by type.



What to do next

You can click the links in the tables to view the performance metrics and usage data in a graph for the monitored Java Virtual Machine.

Related information

[Resource monitoring on page 845](#)

[Viewing resource monitoring usage data on page 885](#)

Monitoring a Docker host

To monitor a Docker host, you must add the Docker host source to HCL DevOps Test Hub (Test Hub), enter connection settings and select the counters that are used to capture the monitoring usage data.

Before you begin

You must have completed the following tasks:

- Logged in Test Hub and you must be the owner of an existing new project.
- Installed a Docker agent. A resource monitoring agent is mandatory to get a Docker data collector. [Resource monitoring agents on page 864](#).
- Started the Docker server to connect the Docker host to Test Hub through the agent.

About this task

The Docker host source is automatically added to the list of resource monitoring sources when the agent is installed.

1. Click **Infrastructure > Resource Monitoring**.
The resource monitoring page opens.
2. In the resource monitoring page, click **Add a Source** and select **add Docker host**.
3. Select an address to connect to the target host in **Access target from**.
4. Click **Add**.
5. Select and save the statistics counters to monitor the source. You can select them from the list where they follow the server logical organization.

For a faster selection, select the counters from the built-in sets drop-down list where they are organized by theme and save your selection. You can select different sets of counters:

- The first level of counters contain generic counters that are related to a Docker Image and all its running Docker Containers (in an exit, running or paused, created, restarted, removing, dead or transitive state).
- The second level of counters contains specific counters that are related to the existing Docker containers.

Result

When the selected counters are saved, two tables are displayed in the resource monitoring main page. They contain the total number of sources you have added and the number of sources ordered by type.



What to do next

You can click the links in the tables to view the performance metrics and usage data in a graph for the Docker host source.

Related information

[Resource monitoring on page 845](#)

[Viewing resource monitoring usage data on page 885](#)

Monitoring a Windows Performance host

To monitor a Windows Performance host, you must add the Windows Performance host source to the project in HCL DevOps Test Hub (Test Hub), and then select the counters that are used to capture the monitoring usage data.

Before you begin

You must have completed the following tasks:

- Installed and started an agent. See [Resource monitoring agents on page 864](#).
- Logged in to Test Hub. You must be the owner of an existing new project.

About this task

The Windows Performance host source is added to the list of resource monitoring sources when the agent is installed.

1. Click **Infrastructure > Resource Monitoring**.
The resource monitoring page opens.
2. In the resource monitoring page, click **Add a Source** and select **add a Windows Performance host**.

3. Select an address to connect to the target host in **Access target from**.

4. Click **Add**.

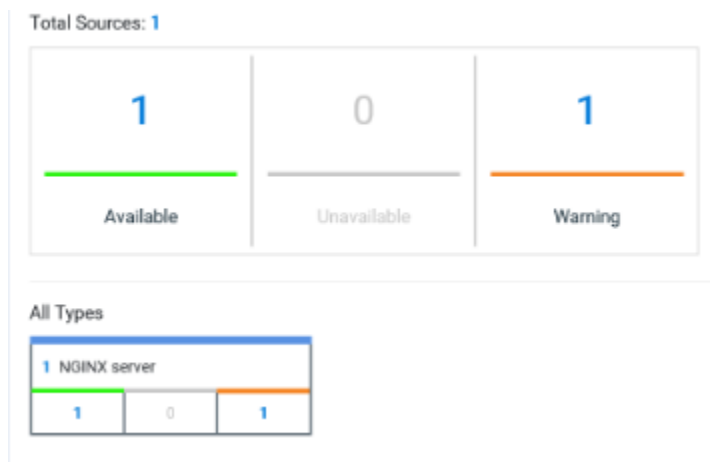
The **Choose Data Counters** dialog opens.

5. Select and save the statistics counters to monitor the source. You can select them from the list where they follow the server logical organization.

For a faster selection, select the counters from the built-in sets drop-down list where they are organized by theme and save your selection.

Result

When the selected counters are saved, two tables are displayed in the resource monitoring main page. They contain the total number of sources you have added and the number of sources ordered by type.



What to do next

You can click the links in the tables to view the performance metrics and usage data in a graph for the monitored Windows Performance host.

Related information

[Resource monitoring on page 845](#)

[Viewing resource monitoring usage data on page 885](#)

Monitoring a Linux Performance host

To monitor a Linux Performance host, you must add the Linux Performance host source to the project in HCL DevOps Test Hub (Test Hub), and then select the counters that are used to capture the monitoring usage data.

Before you begin

You must have completed the following tasks:

- Installed and started an agent. See [Resource monitoring agents on page 864](#).
- Logged in to Test Hub. You must be the owner of an existing new project.

About this task

The Linux Performance host source is added to the list of resource monitoring sources when the agent is installed.

1. Click **Infrastructure > Resource Monitoring**.

The resource monitoring page opens.

2. In the resource monitoring page, click **Add a Source** and select **add a Linux Performance host**.

3. Select an address to connect to the target host in **Access target from**.

4. Click **Add**.

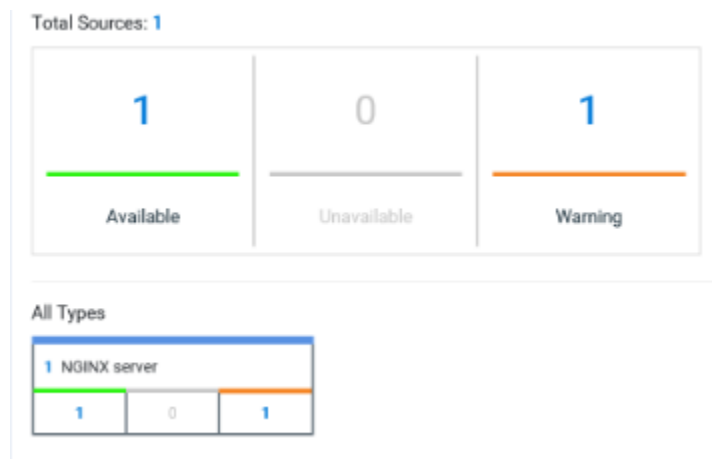
The **Choose Data Counters** dialog opens.

5. Select and save the statistics counters to monitor the source. You can select them from the list where they follow the server logical organization.

For a faster selection, select the counters from the built-in sets drop-down list where they are organized by theme and save your selection.

Result

When the selected counters are saved, two tables are displayed in the resource monitoring main page. They contain the total number of sources you have added and the number of sources ordered by type.



What to do next

You can click the links in the tables to view the performance metrics and usage data in a graph for the monitored Linux Performance host.

Related information

[Resource monitoring on page 845](#)

[Viewing resource monitoring usage data on page 885](#)

Resource monitoring agents

You can connect agents to HCL DevOps Test Hub (Test Hub) from other hosts to monitor a larger set of data collectors, computers, servers, or systems, or if a server to monitor requires authorization access. To monitor these remote servers, you must configure and run the agents so that a connection is set between Test Hub resource monitoring service and the agents. You can configure the resource monitoring agents in a team space or in a project.

Requirements:

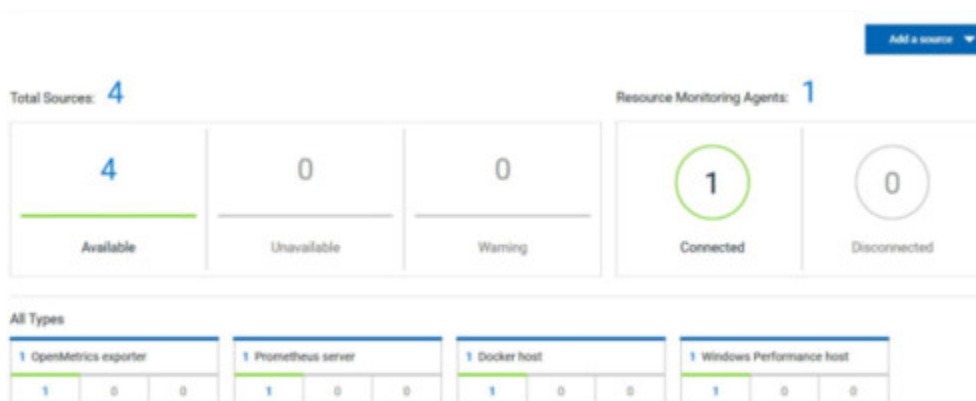
- To connect resource monitoring agents to Test Hub in a team space, you must be a *Team Space Owner*. Any other team space *Member* (not owner) can only view the resource monitoring agents.
- To connect resource monitoring agents to Test Hub in a project, you must be a project *Owner*. Any project *Member* (not the owner) can only view the resource monitoring agents.
- Agent-based monitoring is mandatory to collect data from a Windows or Linux Performance host or from a Docker host.

For testing, you might need many hosts. For example, you might have one host with the application server, another host with the database server, and some hosts to apply the user load. Due to network or firewall issues, sometimes, it becomes difficult for multiple hosts to connect to each other. Resource monitoring agents are installed on the target hosts so that they can establish a connection with Test Hub to gather resource statistics of the target host.

The agents always try to connect with the server through the HTTPS protocol. You must install the agent and start it. From the resource monitoring page in Test Hub, you can copy the command lines to download files and run the agents. When you stop the agent, the monitoring sources that you have already added persist but the live data will not be available.

You can configure your resource monitoring agent as a service so that the agent automatically runs when the host is restarted. For details, see the documentation pages about starting resource monitoring agents on Windows or Linux.

When the agent is connected to Test Hub, it is added to the main page of the resource monitoring service. Then, you can select the agent to connect to a server from **Add a Source**.

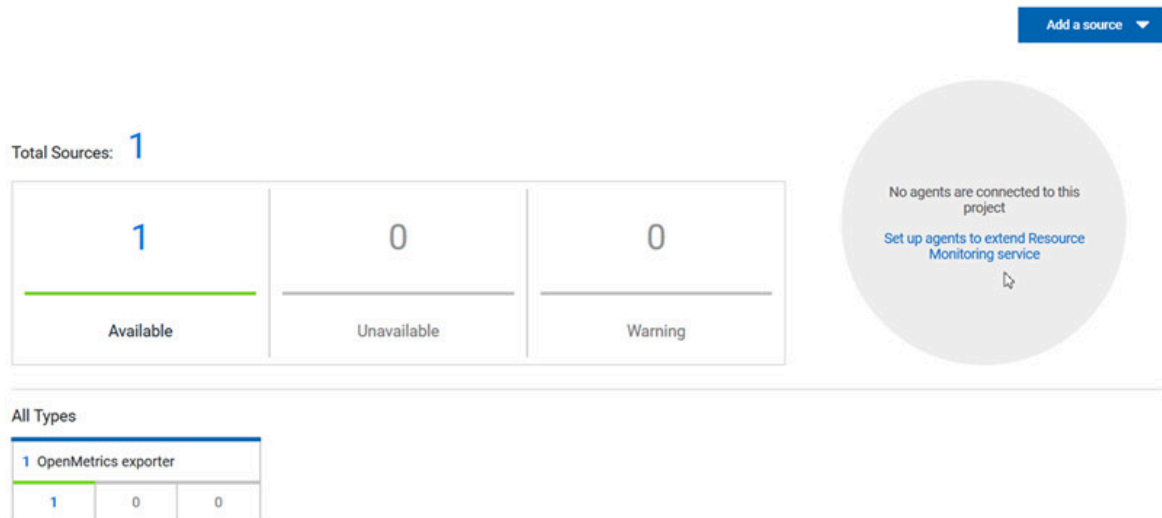


Installing and running a Docker agent on a remote host

You must install the resource monitoring Docker agents on the target host for which you want to monitor the resources and collect the metrics. You need to run an agent to establish a connection with HCL DevOps Test Hub (Test Hub).

Before you begin

- The resource monitoring service does not require access to the agent host but the agent must have reached the service host over HTTPS.
 - The Docker agent must have been launched in a Docker container and requires Docker 19.03.
 - Read and been familiar with the requirements to install and configure the resource monitoring agents on Test Hub. See [Resource monitoring agents on page 864](#).
1. Click **Set up Agents to extend Resource Monitoring service** link in Test Hub to access the agents page where you can find the instructions and commands that are to be used to install and run the agents.



Extend the Resource Monitoring service with agents

▶ Context

▶ Configure the Java agent

▼ Configure the Docker agent

1. Install the Docker agent

[Download Dockerfile](#)

Alternatively, you can use one of the following commands to download the Dockerfile resource:

```
curl -O -J https://tp-cicd6.nonprod.hclpnp.com:/rm/agent-Dockerfile
```

Or:

```
wget --content-disposition https://tp-cicd6.nonprod.hclpnp.com:/rm/agent-Dockerfile
```

2. Build the Docker image

Once you have downloaded the Dockerfile on your host system, launch the following command to build a dedicated Resource Monitoring Docker image:

```
docker build --network=host --rm -t rmagent .
```

3. Run the Docker container

- Expand **Configure the Docker agent** and click **Download Dockerfile** to download a Docker agent. You can also use the `curl` or `wget` commands to download the agent without accessing the resource monitoring web UI.

For more facility, use the code snippets to copy and fill in the commands with the valid offline token and the Docker file name.

For Windows:

```
curl -O -J https://hostName/rm/agent-Dockerfile
```

For Linux:

```
wget --content-disposition hostName/rm/agent-Dockerfile
```

- Copy the build command that is under **Build the Docker image**, paste it in your console and run it to build a resource monitoring Docker image.

Before you start the a Docker agent, you must build a Docker image.

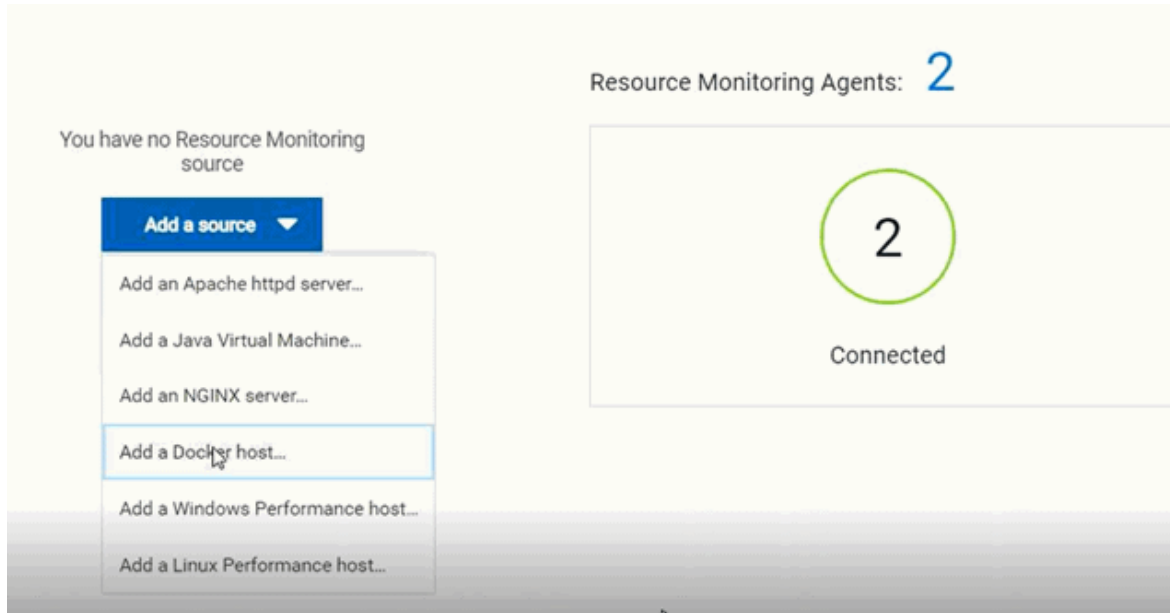
```
docker build --network=host --rm -t rmagent
```

- Follow these steps to run the agent:
 - Click  to fill in the appropriate command for Windows or Linux with the valid offline token.
 - Click  to copy the command to clipboard and paste the command in the console.
 - Command on Windows:


```
docker run -it --rm --privileged -e PROJECT_ID=230200 -e HCL_ONETEST_OFFLINE_TOKEN=(Enter your offline token here) --network host -v //var/run/docker.sock:/var/run/docker.sock rmagent
```
 - Command on Linux:

```
docker run -it --rm --privileged -e PROJECT_ID=230200 -e HCL_ONETEST_OFFLINE_TOKEN=(Enter your offline token here) --network host -v /var/run/docker.sock:/var/run/docker.sock -v /etc:/etc -v /proc:/proc rmagent
```

c. Paste the command in the console and run the command.



Results

When the agent is started, the agent is connected to Test Hub. The agent is displayed in the list of connected resource monitoring agents with the hostname and status of the agent. The agent is also added to the main page of the resource monitoring service in Test Hub.

What to do next

You can add a new resource monitoring source. It can be collected from a service or from the named agents, depending on the capabilities supported by the environment.

Related information

[Resource monitoring capabilities on page 846](#)

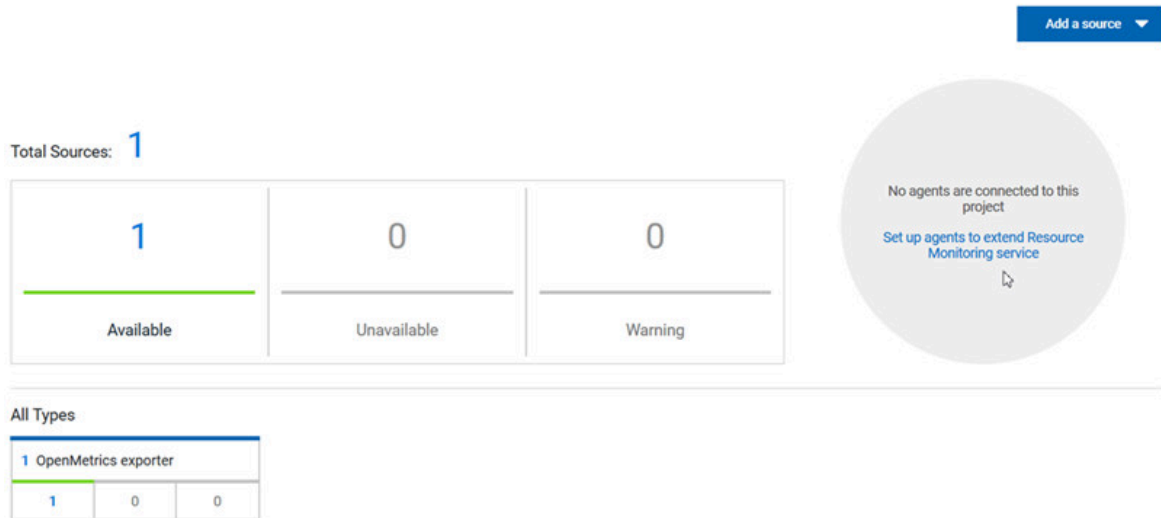
Installing and running a Java agent on a remote host

You must install the resource monitoring java agent on the target host for which you want to monitor the resources and collect the metrics of HCL DevOps Test Hub (Test Hub). You need to run the agent to establish a connection with Test Hub.

Before you begin

- The resource monitoring service does not require access to the agent host but the agent must have reached the service host over HTTPS.
- The Java agent must have been launched from a jar file and requires a Java 17 virtual machine.
- Read and been familiar with the requirements to install and configure the resource monitoring agents on Test Hub. See [Resource monitoring agents on page 864](#).

1. Click **Set up Agents to extend Resource Monitoring service** link in Test Hub to access the agents page where you can find the instructions and commands that you can use to install and run the agents.



2. Expand **Configure the Java agent** and click **Download jar file** to download the Java agent. You can also use the *curl* and *wget* commands to download the agent without accessing the resource monitoring web UI. For more facility, use the code snippets to copy and fill in the commands with the valid offline token and the jar file name.

With curl:

```
curl -O -J https://hostName/rm/Agent-jar
```

With wget:

```
wget --content-disposition https://hostName/rm/Agent-jar
```

3. Configure the command as follows to run your agent:
 - a. Copy the appropriate command for Windows or Linux with the code snippets.
 - b. Enter the path to the directory that contains the agent .jar file, and paste the command.

Example

To connect resource monitoring agents to Test Hub in a team space use the following commands:

On Windows:

```
set HCL_ONETEST_OFFLINE_TOKEN=(Enter your offline token here)
java -jar (Enter the name of the downloaded jar file here) --ServiceUrl=https://hostName/rm
--spaceId=<team-space_id>
```


On Linux:

```
sudo HCL_ONETEST_OFFLINE_TOKEN=(Enter your offline token here) java -jar (Enter the name of the
downloaded jar file here) --ServiceUrl=portNumber/rm --spaceId=<team-space_id>
```

Example

To connect resource monitoring agents to Test Hub in a project use the following commands:

On Windows:

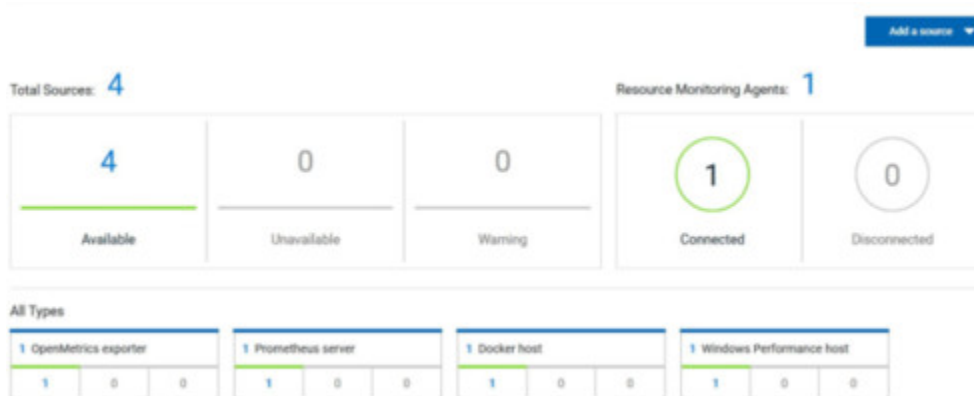
```
set HCL_ONETEST_OFFLINE_TOKEN=(Enter your offline token here)
java -jar (Enter the name of the downloaded jar file here) --ServiceUrl=https://hostName/rm
--projectId=<project_id>
```

On Linux:

```
sudo HCL_ONETEST_OFFLINE_TOKEN=(Enter your offline token here) java -jar (Enter the name of the
downloaded jar file here) --ServiceUrl=portNumber/rm --projectId=<project_id>
```

Results

When the agent is started, the agent is connected to Test Hub. The agent is displayed in the list of connected resource monitoring agents with the hostname and status of the agent. The agent is also added to the main page of the resource monitoring service in Test Hub.



What to do next

You can use another method to start an agent. See [Starting a resource monitoring Java agent as a service on Windows on page 870](#) or [Starting a resource monitoring Java agent as a service on Linux on page 873](#). You can then add a new resource monitoring source. Resource monitoring metrics can be collected from a service or from the named agents, depending on the capabilities supported by the environment.

Related information

[Resource monitoring capabilities on page 846](#)

Starting a resource monitoring Java agent as a service on Windows

To ensure that the resource monitoring Java agent starts by itself when the host is running, you can set up the environment in such a way that the resource monitoring agent can be started as a service.

Before you begin

You must have completed the following tasks:

- Installed Java 17 on the host.
- Added HCL DevOps Test Hub (Test Hub) to the PATH environment variable.
- Created an offline token to connect the agent securely with appropriate permissions.



Note: You can create an offline token from the User menu of the resource monitoring page or you can re-use your active offline token. The token expires if it is not used for a month.

To allow remote access to the performance data on Windows 10, you must have considered the following information:

- The username and password are the same on the local and remote servers. Otherwise you must have provided remote server credentials.
- The remote user must be a member of the Performance Monitor Users group (start `lusrmgr.msc` and add the user to this group).
- The Remote Registry service must be running on the remote host (start `services.msc`) and must have verified the Remote Registry status).
- File and printer sharing must be enabled on the Network Interface of the remote host that is implied in the communication with the local host.
- You must have activated the following Windows Firewall rules so that the remote firewall does not block the access:
 - File and Printer Sharing (NB-Name-In)
 - File and Printer Sharing (NB-Session-In)
 - File and Printer Sharing (*)

1. Download the latest release of `winsw`.

You can choose to download `WinSW.Net2.exe` or `WinSW.Net4.exe` depending on the version of .Net framework that you already have on the host Windows.

2. Create a folder on your local hard drive like `RMAgent-winservice`.

3. Copy the downloaded executable to this new folder and rename the file to `RMAgent-winservice.exe`.

4. Create a new text file in the same folder and name it `RMAgent-winservice.xml`.

5. Copy and adapt the following content to this new `RMAgent-winservice.xml` file to set up the offline token:

```
<service>
  <id>RMAgent-winservice</id>
  <name>Resource monitoring agent</name>
  <description>This service runs the resource monitoring agent.</description>
```

```

<executable>java</executable>
<env name="HCL_ONETEST_OFFLINE_TOKEN" value="(Enter your offline token here)"/>

<arguments>-jar %BASE%\RMAgent.jar --serviceUrl=https://<service-host>/rm
--projectId=<projet-id>
--autoUpgradeDownloadThen=execute:cmd,/c,start,%BASE%\auto-upgrade.bat</arguments>
<onfailure action="restart" delay="10 sec"/>
<logmode>rotate</logmode>
</service>

```

**Notes:**

- Replace `<service-host>` by the hostname of the host that runs Test Hub.
- Replace `<project-id>` which is the number that you find after `/projects/` in the browser's URL when you browse to this project.

6. Create a new text file in the same folder and name it `auto-upgrade.bat`.

7. Copy the following content to this new file to upgrade the agent `.jar` file automatically:

```

@echo off
for /f "tokens=*" %%a in ('dir /b /od %BASE%\com.hcl.test.rm.agent-*.jar') do set newest=%%a
%BASE%\RMAgent-winservice.exe stop
del %BASE%\RMAgent.jar
mklink %BASE%\RMAgent.jar %BASE%\%newest%
%BASE%\RMAgent-winservice.exe start

```

8. Start the command prompt as an administrator and change the directory to the newly created directory:

```
RMAgent-winservice.
```

9. Download the agent `.jar` file that is available from the agents page in the resource monitoring service, under the **Extend the Resource Monitoring service with agents** section and save it to the same directory.

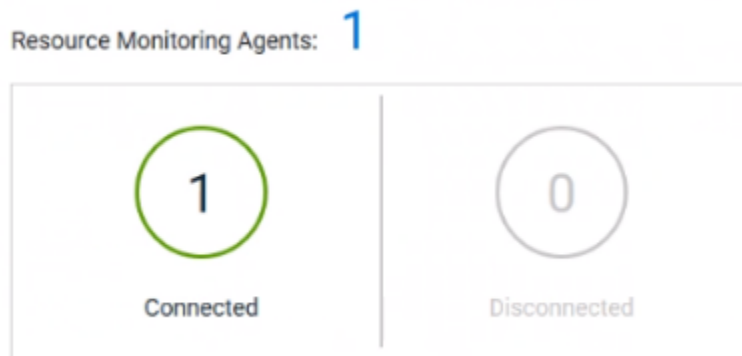
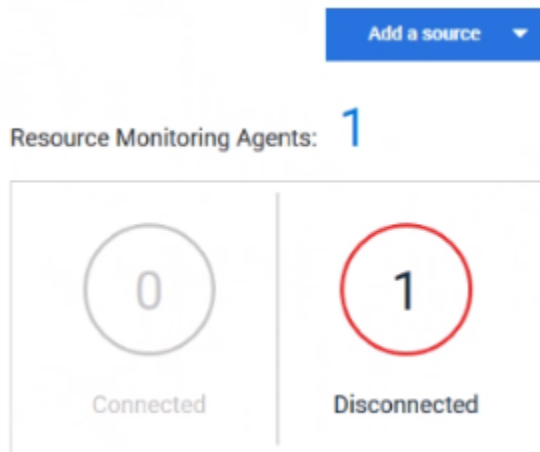
10. Enter the following command in the command prompt to create a symbolic link named `RMAgent.jar` to the agent `jar` file:

```
mklink RMAgent.jar com.hcl.test.rm.agent-<version-and-datetime>.jar
```

11. Enter the following commands:

```
RMAgent-winservice install
```

```
RMAgent-winservice start
```



 **Notes:**

In the Services Windows application, you can check whether the service is up and running. You can see in the resource monitoring page whether the resource monitoring agent is disconnected or connected.

The resource monitoring agents write the logs to the same folder in files named `RMAgent-winservice.out.log`, `RMAgent-winservice.err.log` and `RMAgent-winservice.wrapper.log`. If the Windows service for the agent is not started or if it is not connected to the resource monitoring service, verify the log files.

Results

The Java agent starts as a service on Windows. When the agent is connected to Test Hub, it is automatically added to the list of resource monitoring sources in Test Hub.

What to do next

You can monitor the Windows agent by selecting counters. See [Monitoring a Windows Performance host on page 861](#).

Related information

[Other commands](#)

[winsw file](#)

[Resource monitoring capabilities on page 846](#)

Starting a resource monitoring Java agent as a service on Linux

To ensure that the resource monitoring Java agent starts by itself when the host is restarted, you can set up the environment in such a way that the resource monitoring agent can be started as a service.

Before you begin

You must have completed the following tasks:

- Installed Java 17 on the host.
- Added HCL DevOps Test Hub (Test Hub) to the PATH environment variable.
- Created an offline token to connect the agent securely with appropriate permissions.



Note: You can create an offline token from the User menu of the resource monitoring page or you can re-use your active offline token. The token expires if it is not used for a month.

About this task

This topic relies on systemd services that are the default on most modern Linux distributions. Other ways may require adaptations of the instructions but the provided script will be a good basis in most cases.

1. Create a folder on your local hard drive like `/opt/RMAgent-linuxservice`.
2. Download the agent .jar file that is available from the Agents page in the resource monitoring service, below the **Extend the Resource Monitoring service with agents** section and save it to the same directory.
3. Create a new file `/etc/systemd/system/RMAgent-linuxservice.service`.
4. Add the following content to this new file:

```
[Unit]
Description = Resource monitoring agent
After = network.target

[Service]
Type = forking
ExecStart = /opt/RMAgent-linuxservice/RMAgent-linuxservice.sh start
ExecStop = /opt/RMAgent-linuxservice/RMAgent-linuxservice.sh stop
Restart = on-failure
RestartSec = 10

[Install]
WantedBy = multi-user.target
```

5. Create another file /opt/RMAGENT-linuxservice/RMAGENT-linuxservice.sh.

```
#!/bin/sh

#!/bin/sh

# Update the 3 following variables with the Server's hostname, project id and offline token:
SERVICE_URL=https://<hostname>/rm
PROJECT_ID=<project-id>
export HCL_ONETEST_OFFLINE_TOKEN=<offline-token>

ARGS="--serviceUrl=$SERVICE_URL --projectId=$PROJECT_ID --autoUpgradeDownloadThen=exitFailure"
SCRIPT=$(readlink -f "$0")
RMAGENT_HOME=$(dirname "$SCRIPT")

# Ensure we're using the latest downloaded jar file
PATH_TO_JAR=`ls -t $RMAGENT_HOME/com.hcl.test.rm.agent-*.jar | head -1`
if [ -z "$PATH_TO_JAR" ]
then
cd $RMAGENT_HOME && { curl -k -O -J $SERVICE_URL/agent-jar; cd -; }
PATH_TO_JAR=`ls -t $RMAGENT_HOME/com.hcl.test.rm.agent-*.jar | head -1`
if [ -z "$PATH_TO_JAR" ]
then
echo "Start the server at $SERVICE_URL to allow download of the latest agent jar file"
echo "Exiting..."
exit 1
fi
fi
SERVICE_NAME="Resource monitoring agent"
#Pid file will reside in this script's folder
PATH_TO_PID=$RMAGENT_HOME/RMAGENT-pid
#Log file will reside in this script's folder
PATH_TO_LOG=$RMAGENT_HOME/RMAGENT.log

case $1 in
start)
echo "Starting $SERVICE_NAME ..."
if [ ! -f $PATH_TO_PID ]; then
nohup java -jar $PATH_TO_JAR $ARGS >> $PATH_TO_LOG 2>&1 &
echo $! > $PATH_TO_PID
echo "$SERVICE_NAME started ..."
else
echo "$SERVICE_NAME is already running ..."
fi
;;
stop)
if [ -f $PATH_TO_PID ]; then
PID=$(cat $PATH_TO_PID);
echo "$SERVICE_NAME stopping ..."
kill $PID;
echo "$SERVICE_NAME stopped ..."
rm $PATH_TO_PID
else
echo "$SERVICE_NAME is not running ..."
fi
;;
restart)
```

```

if [ -f $PATH_TO_PID ]; then
    PID=$(cat $PATH_TO_PID);
    echo "$SERVICE_NAME stopping ...";
    kill $PID;
    echo "$SERVICE_NAME stopped ...";
    rm $PATH_TO_PID
    echo "$SERVICE_NAME starting ..."
    nohup java -jar $PATH_TO_JAR $ARGS >> $PATH_TO_LOG 2>&1 &
        echo $! > $PATH_TO_PID
    echo "$SERVICE_NAME started ..."
else
    echo "$SERVICE_NAME is not running ..."
fi
;;
esac

```

**Notes:**

- Replace `<service-hostname>` by the hostname of the host that runs Test Hub.
- Replace `<project-id>` which is the number you'll find after `/projects/` in the browser's URL when browsing to this project.

Results

The resource monitoring Java agent starts automatically when the host restarts. The Linux agent is added to the main page of the resource monitoring service in Test Hub.

What to do next

You can monitor the Linux Performance host source by adding counters. See [Monitoring a Linux Performance host on page 862](#).

Related information

[systemd services](#)

[Resource monitoring capabilities on page 846](#)

Starting a Java Virtual Machine

To monitor the sources from a Java Virtual Machine (JVM), you must first start the JVM. You must enter the IP address of the JVM, the IP port, and the security parameters in the command before running the Java Virtual Machine.

1. Enter The IP address of the JVM (local or remote host) and the IP port in the command that is used to run the Java Virtual Machine.
2. **Optional:** You can use the authentication security data to start the virtual machine. In this case, you must enter the name of the password file. You can also enter the name of an access file that might be needed if user privileges are required.

Example

Parameters used to launch a JVM without security:

```
java
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=9010
-Dcom.sun.management.jmxremote.local.only=false
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
-jar MyapplicationFile.jar
```

Parameters used to launch a JVM with authentication security:

```
java
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=9010
-Dcom.sun.management.jmxremote.local.only=false
-Dcom.sun.management.jmxremote.authenticate=true
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.password.file=jmxremote.password
-Dcom.sun.management.jmxremote.access.file=jmxremote.access (this command is optional)
-jar MyapplicationFile.jar
```

Related information

[Resource monitoring agents on page 864](#)

[Monitoring a Java Virtual Machine on page 858](#)

Managing sources


After you add and connect monitoring sources to your project, you might want to modify the sources in you project and the configuration of the sources.

Before you begin


You must be a project owner or team space owner to add or delete sources.

About this task



In the page where you can view the resource monitoring usage data and metrics, you can also manage your sources. You can modify or delete a source, add or change statistic counters, add labels to identify the sources and use them when you execute a test to collect resource monitoring results.

1. Click **Infrastructure > Resource Monitoring** in the navigation pane.
If you have sources and agents in your project, you can see the cards with the number of available sources and agents in your project.
2. Click a link in the card of available sources to open a list of sources.
3. Perform any of the following actions for a monitoring source:
 - Click the **Delete** icon  to delete the monitoring source and click **Delete** to confirm the deletion in the dialog box.

The source is removed from the list of sources.

- Click the **Edit** icon  to change the target host settings.

You can change the target host address, the port, the path to the server status page and the security parameters.

- Click the **Selection** icon  to select additional counters or change the selected counters for other ones. The changes are reflected in the resource monitoring usage data graph.
- Click  to add a label to identify the source.
- Click **Add a source** to add a source to your project from the list of sources.

Related information

[Resource monitoring capabilities on page 846](#)

[Viewing resource monitoring usage data on page 885](#)

Creating labels


After you connect a host or service to a project, you can add monitoring labels to identify the monitoring sources that are used when you execute a schedule from HCL DevOps Test Hub (Test Hub). You can also use labels in any test to collect resource monitoring usage data during a test run for a defined period of time.

Before you begin

You must have connected sources to your project in Test Hub. See [Resource monitoring capabilities on page 846](#).

About this task

You must be a project owner to add labels to the resource monitoring sources in a project and you must be a team space owner to add labels in a team space.

1. Click a source that is available in your project, either in the **Total Sources** cards or in the **All Types** cards from the resource monitoring page.
2. Click  under the **Labels** column of a resource monitoring source to add a label.
3. Enter a name for the label and press the Enter key.



Note: You can add one label or multiple labels for the same resource monitoring source. A label can contain a maximum of 200 characters.



Tip: You can create the same label for different sources that belong to the same logical group to monitor the whole group with only one label.

Results

The host source is identified by a label.

What to do next

You can use the created labels in tests to collect resource monitoring results or persistent usage data. See [Controlling resource monitoring sources in a schedule on page 878](#) and [Collecting and viewing resource monitoring usage data on page 883](#).

Controlling resource monitoring sources in a schedule

In HCL DevOps Test Hub (Test Hub), you can control the resource monitoring sources that are collected in a performance schedule and override them with the labeled resource monitoring sources that are available in your current Server project.

Before you begin

- You must have performed the following tasks in HCL DevOps Test Performance (Test Performance):
 - Created a performance schedule of **Rate** or **VU** type.
 - Added resource monitoring sources, or the labels of resource monitoring sources to a schedule. See [Adding resource monitoring sources to a performance schedule by using labels](#).
 - Enabled Test Hub as resource monitoring service only if you want to replace resource monitoring labels set in Test Performance with the labels that you created in Test Hub.
- You must have created a project and added resource monitoring sources to your project in Test Hub. See [Resource monitoring capabilities on page 846](#).

About this task

This procedure describes how to create resource monitoring labels in Test Hub and how to use them to override the source or label settings related to resource monitoring in a performance schedule. The procedure applies to the following use cases:


- To replace the resource monitoring labels that were initially set in a schedule from Test Performance with the labels that you create in Test Hub.
- To enable resource monitoring sources that were not enabled from Test Performance. In that case, you can enable resource monitoring sources from the **Execution** dialog in Test Hub.
- To ignore resource monitoring sources that were set in the schedule and to change them for labels that match sources that are available in the Test Hub project.



Notes: You must be a project owner to add labels to the resource monitoring sources in a project and you must be a team space owner to add labels in a team space. You must have a project owner or a tester role access to add labels to a performance schedule in a project.

1. Click a source that is available in your project, either in the **Total Sources** cards or in the **All Types** cards from the resource monitoring page.



2. Click  under the **Labels** column of a resource monitoring source to add a label.

The screenshot shows the source list table with the following columns: Name, Status, Added, Observed, and Labels. The 'Labels' column contains a plus icon for each source, which is highlighted in blue in the image.

Name	Status	Added	Observed	Labels
> OpenMetrics at to: [redacted]	available	7 hours ago	6 hours ago	[SUTv2] +
> Prometheus at [redacted]	available	7 hours ago	29 minutes ago	[SUTv2] +

3. Enter a name for the label and press the Enter key.


Example: Enter 'SUTv2' in three different resource monitoring sources.



Note: You can add one label or multiple labels for the same resource monitoring source. A label can contain a maximum of 200 characters.



Tip: You can create the same label for different sources that belong to the same logical group to monitor the whole group with only one label.

4. Click **Execution** in the navigation pane and click  on a performance schedule in the list of tests to execute a schedule.
5. Click the **Resource Monitoring** tab in the **Execute Test Asset** dialog.

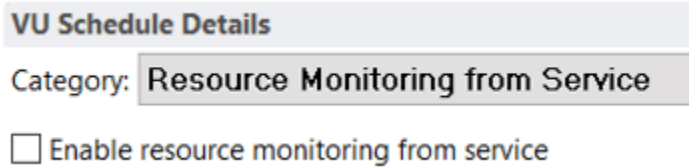


Note: There are different use case scenarios:



- If the **Resource Monitoring from Service** option was not enabled in Test Performance (see Figure 1), no source is available in Test Hub project.

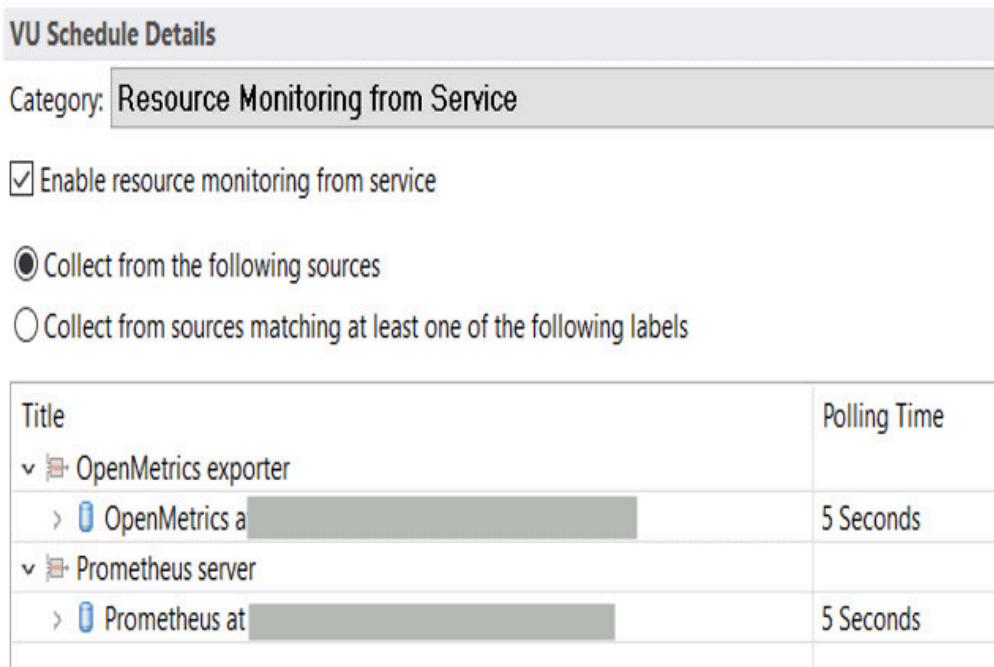
Figure 1:



A message indicates in the **Resource Monitoring** tab that Resource Monitoring was not enabled. You can run the schedule with the resource monitoring sources that are created and labeled in Test Hub (steps 1 to 6 in this procedure) by selecting labels that are in your Test Hub project (step 6).

- If the **Resource Monitoring from Service** option was enabled in Test Performance, and data is collected from the sources you added to the performance schedule in Test Performance (see Figure 2), the **Resource Monitoring** tab displays the list of these sources that are available in your Test Hub project.

Figure 2:



To override the initial sources with the sources that you created in Test Hub, select the labels that are in your Test Hub project and that match the Server sources (step 6 in this procedure).

- If you added sources to your performance schedule by using labels in Test Performance (see Figure 3), the **Resource Monitoring** tab displays the list of labels that are available in your Test Hub project.



Figure 3:

VU Schedule Details

Category: **Resource Monitoring from Service**

Enable resource monitoring from service

Collect from the following sources

Collect from sources matching at least one of the following labels


OpenM	Add...
Prom	Edit...
SUTv2	Remove

To select resource monitoring labels from your schedule, follow step 6 in this procedure.

In the list of sources or labels that is displayed in the **Resource Monitoring** tab:

- Green icons identify either labels that match the sources or sources that are found in the current project.
- Red icons indicate that the sources are not running,
- Yellow icons identify either labels that do not match the sources, or sources that are not found in the current project.

Data is collected for the sources that match labels only.

6. Click the , and press the `Ctrl + Space` keys, or enter the initial letter of a label to select a label in the list.

Example: Select **SUTv3**.

ENVIRONMENT DATA SOURCES VARIABLES LOCATION **RESOURCE MONITORING**

Asset settings: list of labels

- ▲ L1_ITS: not matching
- ▲ L2_Proj_ITS: not matching

To collect data during this execution, you can select existing labels that match sources:


SUTv3 x +

Label selection summary:

- ✔ SUTv3: matching
 - ✔ Prometheus at tp-junit1.nonprod.hclpnp.com:48090 Defined in team space
 - ✔ OpenMetrics at tp-junit1.nonprod.hclpnp.com:48080 Defined in team space

Advanced Reset Cancel Execute

You can see all the labels that you added to the resource monitoring sources in Test Hub.

 **Note:** You can select labels matching resource monitoring sources that are created in a team space in addition to the ones that are created in a project.

7. Select the labels that match the sources.
8. Click **Execute** to execute the performance schedule.
9. Click the **Result** view in the navigation bar of Test Hub when the schedule is complete.
10. Expand the performance schedule result to see the resource monitoring label name in the **Details** card.

The **Details** card displays the resource monitoring labels that are used in the schedule to override the initial labels and collect resource monitoring sources (in the example, the 'SUTv3' label). For other details about the **Results** page, see the link at the end of this page.

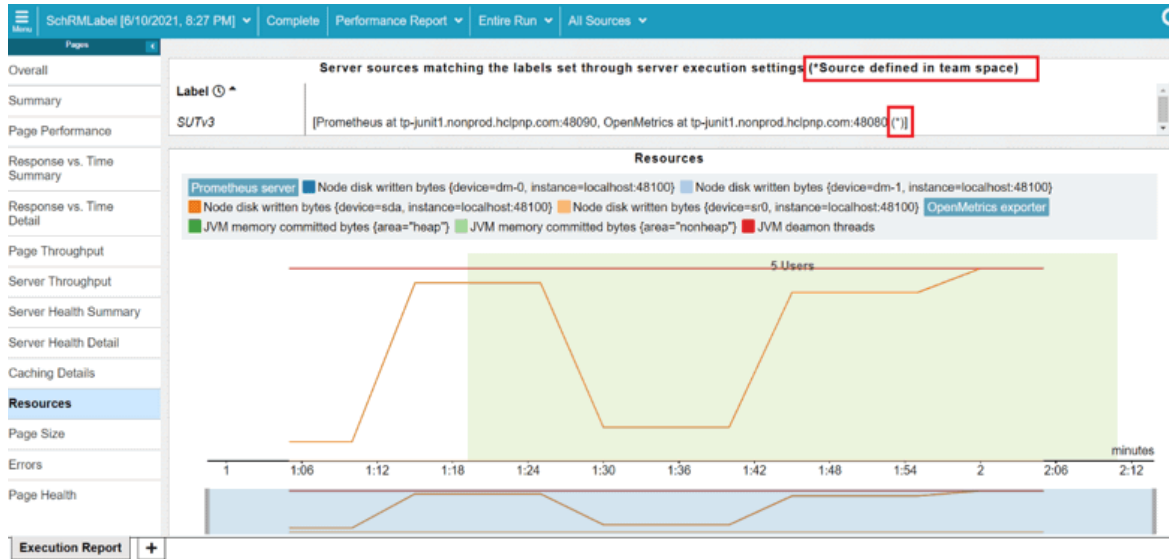


SchedListOfRMSourcesEnabled
proj_RM_1011/Schedules/SchedListOfRMSourcesEnabled.tes... dilip Sep 8, 2020 10:16 AM 00:04:25.153 Pass

DETAILS		Reports	
Id	1751	Statistics Report	
Branch	Dilip_1011	Test Log (Text)	
Result	SchedListOfRMSourcesEnabled		
Finish on	Sep 8, 2020 10:20 AM		
Status	Complete		
Git repo	https://[redacted]testingproducts/PT_SVT_TestAsset.git		
Git path	proj_RM_1011/Schedules/SchedListOfRMSourcesEnabled testsuite		
Git commit	commit sch with RM sources CRC server [redacted] committed 22 days ago (#refs/heads/Dilip_1011) d24e819		
Execution location	Default Cluster		
Resource Monitoring labels (override)	SUTv3		

In the **Report** card, you find the link to the Statistics Report that is generated. In the **Resource** tab of the Statistics Report, you can view the information related to the Server sources matching the labels set in the schedule.

- Click the **Statistics Report** link in the **Reports** card to see the performance metrics of the monitoring sources matching the selected labels.



Example: You can see the statistics for the sources tagged with the 'SUTv3' label.



Note: In the Resources report, each resource name followed by an asterisk identifies a resource monitoring source that is defined in a team space.

Related information

[Test results and reports on page 789](#)

Collecting and viewing resource monitoring usage data

With HCL DevOps Test Hub (Test Hub), you can collect resource monitoring data in a timeframe and view the resource monitoring usage data in a graph while the test is running. You can also create queries to retrieve the resource monitoring usage data.

About this task

While monitoring a system, you can view the current data in a graph but you might also want to view persistent data that is collected during a test run in a time frame. To view the persistent data, you can configure the test with additional parameters before the test execution. While the test is running, you can view the resource monitoring usage data in a graph. When the test is running or complete, you can create queries to retrieve the collected data results.

To collect and view resource monitoring data, perform these tasks:

1. Configure a test with two additional parameters in the advanced settings.
2. View the resource monitoring usage data in a graph.
3. Create queries to retrieve the resource monitoring usage data.

Related information

[Configuring a test on page 884](#)

[Viewing resource monitoring usage data on page 885](#)

[Viewing resource monitoring usage data on Grafana on page 887](#)

[Creating queries for resource monitoring usage data on page 889](#)

Configuring a test

You must configure a test with two parameters to collect resource monitoring usage data in a time frame when the test is running.


Before you begin

You must have completed the following tasks:



- Added and configured a source.
- Created resource monitoring labels for your source.

About this task

You can collect resource monitoring usage data during a test run for all types of monitoring sources, and all test types, except API stubs.

1. Click **Execution**, In the navigation pane.
2. Select the test that you want to run and click  in the **Actions** column to open the **Execute test asset** window.
3. Click **Advanced settings**.
4. Enter the following parameters and values in **Additional Configuration Parameters**:

Parameter name	Value
rm.log.labels	<p>Enter a label name in the second edit field as the parameter value.</p> <p>If you enter multiple labels, enter a comma to separate them in the list.</p>

Parameter name	Value
	<p> Example: For two labels with the following names: 'LabelA' and 'LabelB':</p> <pre>rm.log.labels LabelA,LabelB</pre> <p>If a label contains a comma in its name, you must add two commas in order to collect the resource monitoring usage data.</p> <p> Example: For a label named 'Label1A,Label1B', enter the following parameter and value:</p> <pre>rm.log.labels Label1A,,Label1B</pre>
<p>rm.log.period.seconds</p>	<p>Enter a sampling period in the second edit field as the parameter value. The minimum period is one second.</p>

5. Click **Add** for each parameter.

6. Click **Execute**.

Results

The monitoring data is collected while your test is running. You can see the collected data in a graph of the resource monitoring page, even 3 minutes after the test is complete.

What to do next

You can view the resource usage data on the **Resource Monitoring** page in HCL DevOps Test Hub (Test Hub) and Grafana.

- [Viewing resource monitoring usage data on page 885](#)
- [Viewing resource monitoring usage data on Grafana on page 887](#)

Viewing resource monitoring usage data

After you add and configure monitoring sources in HCL DevOps Test Hub (Test Hub), you can view the resource monitoring usage data in a graph for each monitored source and for each selected counter.

Before you begin

You must have completed the following tasks:

- Added and connected a monitoring source, and an agent when required, to your project to be able to view the list of available sources and agents, and the resource monitoring usage data in your project.
- Configured a test with advanced parameters to view persistent usage data in a graph.

About this task

After you add a monitoring source to Test Hub, configure the source and select monitoring counters, you can view the current resource monitoring usage data for each selected counter in a graph. If you configure the test to collect data, you can also view the resource monitoring usage data that is collected for a defined timeframe. In this case, the data is still visible in the graph three minutes after the test run is complete. After this time, you can create queries to retrieve the data.

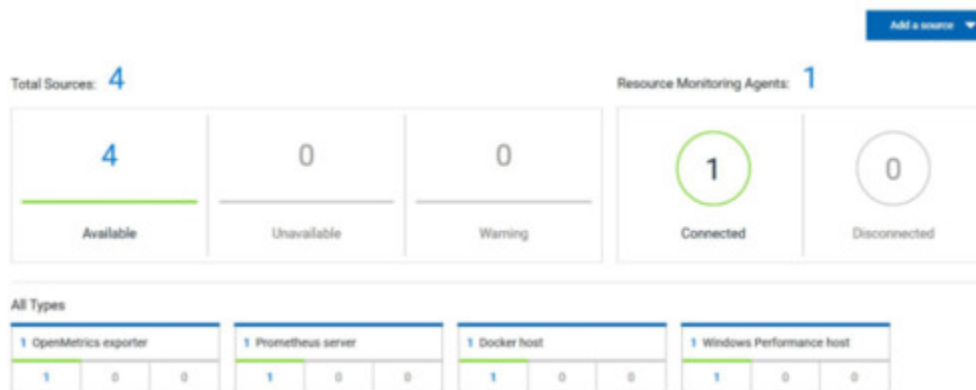


Note: For a Prometheus source, as data comes from the Prometheus database, it no longer shows up in the graph when the test run is complete. However, the data is collected and shows up in the queries if it matches a label.

1. Click a link either in the **Total Sources** cards or in the **All Types** cards from the resource monitoring page to access the resource monitoring usage data.

The tables display the total number of sources in your project and the number of sources by type. You can also view the number of available and unavailable sources, and the number of sources that contain warnings. If you have added '4' sources to your project, for example, click '4' in the table to access the page where you can see the resource usage data for each source.

If you have added agents to your project, you can also view the number of agents that are connected and disconnected to your project.




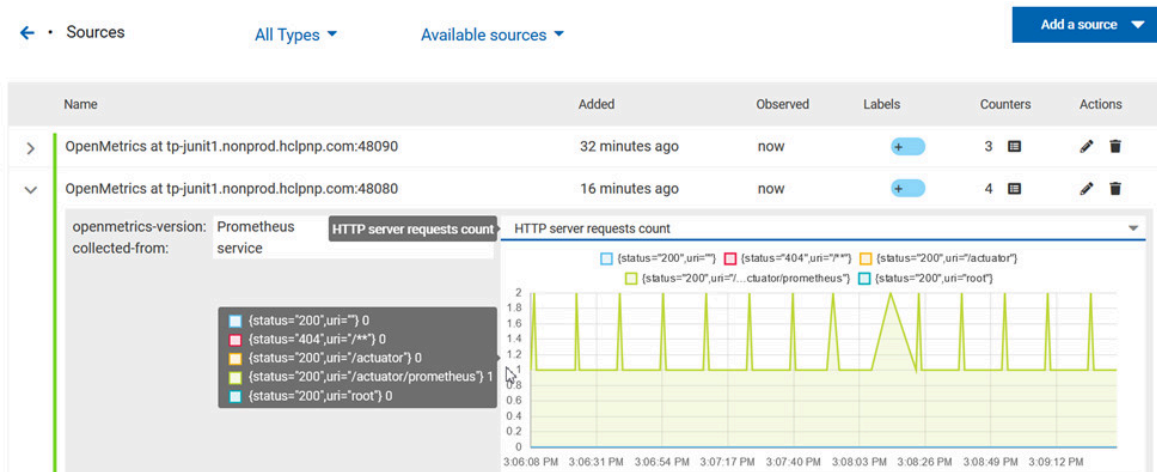
Result

The list of sources is displayed in a new page.

You can filter the list of sources by collector type or by status.

2. Expand a monitoring source under the **Name** column to see the current monitoring data usage in a graph.

3. Click the down arrow  to select a counter in the list.
A graph is displayed for each counter.



What to do next

From the page where you can view the resource monitoring usage data, you can also manage your sources. See [Managing sources on page 876](#).

Related information

[Collecting and viewing resource monitoring usage data on page 883](#)

[Creating queries for resource monitoring usage data on page 889](#)

Viewing resource monitoring usage data on Grafana

After you add and configure monitoring sources in HCL DevOps Test Hub (Test Hub), you can view the resource monitoring usage data in a graph for each monitored source and for each of the selected counters on Grafana.

Before you begin

You must have completed the following tasks:

- Added and connected a monitoring source, and an agent when required, to your project to view the list of available sources and agents, and the resource monitoring usage data in your project.
- Configured a test with advanced parameters to view persistent usage data in a graph. See [Configuring a test on page 884](#).
- Installed the DevOps Test Hub Results Data plug-in and created an instance of data source for DevOps Test Hub Results Data. See [Installing and configuring the DevOps Test Hub Results Data plugin on page 802](#).
- Opened Grafana with one of these ports: `http://localhost:3000` or `http://localhost:8080`.
- Started the test execution on Test Hub.

About this task

After you add a monitoring source to Test Hub, you must configure the source, and select monitoring counters. You can view the resource monitoring usage data for each of the selected counters in a graph on Grafana. You must configure the test to collect and view the resource monitoring usage data that is collected for a defined timeframe. During the test execution, the data can be seen with the default time range on Grafana. After the test execution is complete, you must carefully select the time range so that it corresponds to the test execution start and end time.


1. Click **Explore** in the **Toggle** menu in Grafana.

The **Data query** page with the default data source is displayed.

2. Select **DevOps Test Hub Results Data** from the list of data sources.
3. Select the type of query by performing any of the following actions when you want to view statistics from the test data:

To view	Actions
The variation of a metric over time.	Perform the following steps: <ol style="list-style-type: none"> a. Click the time-series statistics tab icon . b. Go to Step 4 on page 888.
The values of the metric that are computed over the entire time range.	Perform the following steps: <ol style="list-style-type: none"> a. Click the overall statistics tab icon . b. Go to Step 4 on page 888.

4. Define the scope of the query by performing the actions indicated in the following table:

Options	Actions
If you want to use the data from the team space	Perform the following steps: <ol style="list-style-type: none"> a. Select the Team Space option. b. Select the team space from the list of team spaces that are listed. c. Select Resource Monitoring from the list in the Service option. d. Go to Step 5 on page 889.
If you want to use the data from your project	<p> Note: You can select the project as the scope for your query if you want to view any of the supported query types.</p> Perform the following steps: <ol style="list-style-type: none"> a. Select the Project option. b. Select the project from the list of projects that are listed. c. Select the Service option. d. Select Resource Monitoring from the list e. Go to Step 5 on page 889.


5. Optionally, enter a name for the metric that helps you identify it in the **Metric** field.
6. Select a metric from the list of metrics in the row of the **Metrics** option.
7. Select the aggregation factor for the metric. For example, select **Sum** to add all values of the metric.
8. Optionally, click the **Cumulated** toggle button to add the values for the metric cumulatively over time.



Note: The **Cumulated** is set to off. This is the default selection.

9. Add another metric by clicking the **Add metric** icon .



Note: If you added multiple metrics and you want to remove a metric, you can click the **Remove metric** icon  of the metric.

10. Click **Add query** if you want to combine queries from different projects, results ids, or with different properties in the query filters.
11. Click **Run query**.

Result

A graph is displayed for the selected metrics.

Results

You viewed the resource monitoring usage data in a graph for each monitored source and for each of the selected counters on Grafana.

What to do next

You can build a dashboard including graphs, charts, gauges, and tables to get customized visual representations of your resource monitoring usage data. See [Dashboard overview](#).

Creating queries for resource monitoring usage data

After you execute a test to collect resource monitoring usage data, you can create queries in HCL DevOps Test Hub (Test Hub) to retrieve the data.

Before you begin

You must have added resource monitoring sources to a project.

About this task

You can no longer view the monitoring usage data in graphs 3 minutes after completion of the test. However, you can create queries to retrieve the list of activities and events in your tests. You can use the following procedure as an example to create your own queries.

1. Follow these steps to filter the list of activities that are collected for a duration of less than 300 seconds:
 - a. Click **Activities**.
 - b. Enter `'Type is RMsource and duration < 300'` in the **Query filters** field.

- c. Enter 'time ASC' in the **Sort by** field to sort data by ascending order.
- d. Click **Run Query**.
- e. Click **Save as**.
- f. Enter a name to identify the new query and click **Save**.

The list of activities that are collected for a duration of less than 300 seconds is displayed in the query result table, and this list is sorted in an ascending order.

2. Follow these steps to filter the list of observations that are performed during a test execution and defined by the test result identifier:
 - a. Click **Events**.
 - b. Enter the following filters in the **Query filters** field:

```
type is observation and anyParentActivity matches ( anyChildActivity matches (requesterNature = 'test-result' and requesterId=1202 ))
```

- c. Enter 'time ASC, parentActivity.description ASC, specifier ASC' in the **Sort by** field to sort data by ascending order.





Note: You can use the **Ctrl-Space** keyboard shortcut to enter the properties and operators.


- d. Click **Run Query**.
- e. Click **Save as**.
- f. Enter a name to identify the new query, and then click **Save**.

The new query results are displayed in the table of activities and events.

3. Follow these steps to manage your query:

- Click the action menu  that is next to the **Save** button and select **Reset** if you have just modified a query and you want to undo your modification.
- Click the  action menu and select **Delete** to delete the new query from the **Events and activities** queries list.

4. Follow any of these steps to configure the query result table:

- Click the action menu  in a column of the table to change the column display, and then perform any of the following actions:
 - Click **Insert new column**, enter a field name in **Search new column** and click **Add**.
 - Click **Move** to move a column in the table.
 - Click **Remove** to remove a column from the table.
- Follow these steps to add a column at the end of the table:

- a. Click **Add column**.



Result: A window opens.

- b. Enter a field name in **Search new column**.
- c. Click **Add**.

5. Click **Save** to save the configuration of the results table.

Results

You have created two queries in your project to display:

- The list of activities that are collected for a duration of less than 300 seconds.
- The list of observations that are performed during a test execution and defined by the test result identifier.

Related information

[Configuring and running log queries on page 824](#)

Configuration of a change management system

When you use any application to create and track defects (bugs), issues, or other work items, you can configure the application as a change management system on HCL DevOps Test Hub (Test Hub). You can then create defects (bugs), issues, or other work items without the need to open your application.

You use Test Hub to run tests for the software that you develop and view their results. When you discover that you might want to raise defects (bugs), issues, or other types of work items for these tests, you are required to open the application that you use. You can only then, create defects (bugs), issues, or other work items in your application.

When you configure your application as a change management system on Test Hub, you can create defects (bugs), issues, or other work items without the need to open your application.

You can find details in the following table about the usage of the applications that you can configure as a change management system on Test Hub.



Note: You can configure only one application as the change management system on Test Hub.

Issue tracking application	Usage as a change management system	Go to...
Atlassian Jira Software	You can create stories, defects, or tasks as issues in your project in Jira from Test Hub and also launch your project in Jira from Test Hub to track and manage these issues.	Configuration of Atlassian Jira as a change management system on page 892

Issue tracking application	Usage as a change management system	Go to...
Azure DevOps	You can create work items in your project in Azure DevOps from Test Hub and also launch your project in Azure DevOps from Test Hub to track and manage these work items.	Configuration of Azure DevOps as a change management system on page 904
Rational® ClearQuest® or HCL® Compass	You can use Rational® ClearQuest® or HCL® Compass as a change management system in Test Hub. You can also launch Rational® ClearQuest® or HCL® Compass from Test Hub to track and manage change requests and other records.	Configuration of Rational ClearQuest or HCL Compass as a change management system on page 914

Configuration of Atlassian Jira as a change management system

You can use Atlassian Jira as a change management system in HCL DevOps Test Hub (Test Hub) to create issues in the Jira project for the tests that ran in Test Hub. The links to the Jira issues are displayed on the **Results** page in Test Hub, and you can then view and track the Jira issues in Jira.

Prerequisites

Before you configure Jira as a change management system on Test Hub, you must have completed the following tasks:

- Installed Jira Software. See [Supported installations on page 892](#).
- Ensured that you obtained the access and authentication details to Jira from the Jira administrator.
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).

Supported installations

The following types of Jira Software installations are supported that can be configured as a change management system on Test Hub:

- On-premise Jira Software installations.
- On-cloud Jira Software installations.

Task flow

You can follow the tasks that are listed in the following table when you want to use Jira as a change management system on Test Hub. The table also provides you the links to the information about the tasks.

Task	Go to...
Configure Jira as a change management system on Test Hub.	Configuring Jira as a change management system on page 893
Create defects or issues for tests that ran on Test Hub.	Creating Jira issues for the test results on page 895
View details of the Jira issues that you created from Test Hub.	Viewing details of the Jira issues on page 902

You can manage the Jira settings that you configured in Test Hub. You can edit the saved configuration or you can delete the configuration. For more information, see [Managing the Jira configuration on page 903](#).

Configuring Jira as a change management system

You must configure Jira as a change management system in your project in HCL DevOps Test Hub (Test Hub) so that can create issues in the Jira project for the tests that ran.

Before you begin

You must have completed the following tasks:

- Verified that you can access Jira.
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).

About this task

With the introduction of OAuth 2.0 authentication for the connection between Test Hub and Jira, you must be familiar with the creation of Apps that are based on the OAuth 2.0 integration. There can be more than one method in the Atlassian Developer console that you can configure the created App. One such method is adopted in the configuration of an App in Jira.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Integration** in the navigation pane.

The **Integration** page is displayed.

4. Click the **CHANGE MANAGEMENT SYSTEM** tab.

5. Click **Add New**, and then select the **Jira Software** option from the list.

The **Add Jira configuration** dialog is displayed.

6. Perform the following steps in the **Add Jira configuration** dialog:

a. Enter the URL of the Jira server or Jira cloud in the **Jira URL** field, and then click **Next**.

Test Hub provides the *Callback URL* that you must use to generate the Client ID and Client secret for the App in Jira.

b. Click the **Atlassian Developer console** link.

The **Atlassian Developer console** page is displayed.

Perform the following steps on the **Atlassian Developer console** page:

i. Click **Create > OAuth 2.0 integration**.

ii. Enter a name for the App in the **Name** field.

iii. Click the **I agree to be bound** checkbox, and then click **Create**.

The App is created in the Atlassian Developer Console and the details are displayed for the App.

iv. Click **Authorization** in the navigation, and then click **Add** in the row of OAuth 2.0 (3LO).

The OAuth 2.0 configuration page is displayed. You must paste the *Callback URL* provided by Test Hub.

v. Copy the *Callback URL* from the **Add Jira configuration** dialog in Test Hub and then paste it into the **Callback URL** field.

vi. Click **Save changes**.

The App in Jira is configured to authenticate the requests from Test Hub by using the OAuth 2.0 protocols.

vii. Click **Permissions** and then, click the **Configure** option in the row of **Jira API**.



viii. Click **Edit Scopes** on the **Jira API** page, and then select the following scopes for your App that are required for a successful connection with Test Hub:

- View Jira issue data
- Manage Jira global settings
- View user profiles
- Create and manage issues


ix. Click **Save** to save the App settings.

x. Click **Settings** and scroll down the page to view the **Authentication details** panel.

The **Client ID** and **Secret** must be copied from the **Settings** page to the corresponding fields on the **Add Jira configuration** dialog.

- c. Click the **Copy** icon  in the **Client ID** field to copy the key, and then paste the copied key in the **Client ID** field on the **Add Jira configuration** dialog.
 - d. Click the **Copy** icon  in the **Secret** field to copy the key, and then paste the copied key in the **Secret** field on the **Add Jira configuration** dialog.
7. Click **Save** on the **Add Jira configuration** dialog.

You added the URL of the Jira cloud and configured an App in Jira to authenticate the connection with Test Hub by using OAuth 2.0. You can view that the URL of the Jira server or cloud instance is displayed on the **CHANGE MANAGEMENT SYSTEM** tab and the status is displayed as *Project not configured*.

8. Click the **Open action menu** icon , and then click **Configure**.

A page in Atlassian is displayed in a browser that states that the App you created previously is requesting access to your Atlassian account.

9. Read the message that is displayed, and then click **Accept**.

When the connection between Test Hub and the App in Jira is established successfully, the **Configure Jira project** dialog is displayed with the Jira URL that is configured as a change management system in Test Hub.

10. Select the project in Jira from the list under which you want to create defects.
11. Click **Save**.

Results

You configured Jira as a change management system in Test Hub.

What to do next

You can perform the following tasks:

- View the following information on the **CHANGE MANAGEMENT SYSTEM** tab about the Jira configuration:
 - The URL of the Jira server or cloud instance.
 - The project that you selected.
 - The status of the configuration.
- Create issues for the test results that are available in your project in Test Hub. See [Creating Jira issues for the test results on page 895](#).

Creating Jira issues for the test results

You can create Jira issues for the tests that ran in HCL DevOps Test Hub (Test Hub). The issues are created in the project in Jira that you configured in Test Hub.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Configured Jira as a change management system for your project in Test Hub. See [Configuring Jira as a change management system on page 893](#).
- Configured a test run. See [Test run configurations on page 446](#).
- Viewed the results of the completed test runs on the **Results** page. See [Test results on page 788](#).

About this task

When you want to create a Jira issue in Test Hub, the issue types that display in the **Create an issue in Jira** dialog depend on the configuration done by the administrator of Jira.

The default fields that display on the **Create an issue in Jira** dialog are as follows:

- **Issue Type**
- **Summary**
- **Description**
- **Attach Reports**

If you want to view any additional fields in the **Create an issue in Jira** dialog from Test Hub, then you must inform your Jira administrator to configure those additional fields as the required fields in the project in Jira. You can then use those fields in the **Create an issue in Jira** dialog when you create a defect from Test Hub.

You can refer to the **Related information** section to know more about Jira field configuration.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Analyze > Results**.

The **Results** page is displayed and all the test assets that were run earlier are displayed in tabs, as listed in the following table:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.

Tab	Description
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.



Note: You can click the relevant tab to view the tests that were run earlier.

4. Perform any of the following actions to identify the test asset:































- Scroll through the list.













Tip: You can hover over the icon in the **Type** column to know the type of the test asset.



Note: You can also identify the type of the asset from the icon that represents the test type as shown in the following table:

Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			





Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test by entering any text that is contained in the test asset name in the **Search** field.
- Select the time interval within which the test was run.

For example, you can select the *Last 24 hours* option to view the tests that were run in the previous 24 hours.

- Select the range of dates within which the test was run.


You can click the calendar icon  to select the date in the first image as the start date, and then select the date in the second image as the end date.

 **Note:** You can select the range of dates only when you select the time interval as *Custom*.

- Click the **Filter** icon  in the **SUITES** or **TESTS** tab, and then select the test type.

All the tests based on the selected test type are displayed.

For example, you can select **API Suite** in the **SUITES** tab to display all the API suites or you can select **Functional Test** in the **TESTS** tab to display all the functional tests.

- Click the **Filter** icon  in the **ADVANCED** tab, and then perform the following actions to create a filter query:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then select **Add rule** to add a rule or **Add group** to add a group of rules.
 - Select or enter the relevant parameters, and then either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- ID
- Type

- Test
 - Version
 - Result
 - Started on
 - Finish on
 - Duration
 - Status
 - Verdict
 - Labels
 - Locked
 - Repository
 - Branch
 - Text
- e. Click **Save**.

The filter query is saved and all the tests that match the filter criteria are displayed.

- Perform the following actions in the **ADVANCED** tab to retrieve and apply a previously saved filter query:

- a. Click the **Filter** icon .

All the filter queries that you created and saved earlier are displayed.

- b. Click the filter query that you want to apply.


All the tests that match the filter criteria are displayed.


You have identified the test that you want to use.

5. Select the step that depends on the role assigned to you in the project on Test Hub:

- If you are a member with a role assigned as *Owner*, go to [Step 6 on page 899](#).
- If you are a member with a role assigned as *Tester*, and if you did not complete the one-time authorization process, go to [Step 7 on page 900](#).
- If you are a member with a role assigned as *Tester* and completed the one-time authorization process, go to [Step 8 on page 900](#).

6. Perform the following steps if you are assigned the role of a project *Owner*:

- a. Click the **Jira: Create a Jira issue** icon  that is under the **Actions** column in the row of the test.

Alternatively, you can click the test under the **Test** column to view the results panel, and then click the **Jira: Create a Jira issue** icon  that is displayed in the panel.


The **Create an issue in Jira** dialog is displayed.


- b. Go to [Step 9 on page 900](#).

7. Perform the following steps if you are assigned the role of a *Tester* and if you have not yet completed the one-time authorization process:



Note: You must complete the one-time authorization process if you are not a project owner.

- a. Click the **Jira: Create a Jira issue** icon  that is under the **Actions** column in the row of the test.

Alternatively, you can click the test under the **Test** column to view the results panel, and then click the **Jira: Create a Jira issue** icon  that is displayed in the panel.

The **Configure Jira server to authorize the application** dialog is displayed.

- b. Click the URL of the Jira server.
- c. Log in to the Jira server account with your valid credentials.




Note: The **Login** page is displayed only if you are not already logged in to Jira.


- d. Read the message that is displayed, and then click **Allow** to authorize the Jira account.

After the authorization is successful, the **Access Approved** page is displayed with a verification code.

- e. Copy the verification code, and then enter the verification code in the field on the **Configure Jira server to authorize the application** dialog of Test Hub.
- f. Click **Save**.
- g. Go to [Step 8 on page 900](#).

8. Perform the following steps if you are assigned the role of a *Tester* and you have completed the one-time authorization process:

- a. Click the **Jira: Create a Jira issue** icon  that is under the **Actions** column in the row of the test.

Alternatively, you can click the test under the **Test** column to view the results panel, and then click the **Jira: Create a Jira issue** icon  that is displayed in the panel.

The **Create an issue in Jira** dialog is displayed.

- b. Go to [Step 9 on page 900](#).

9. Perform the following steps in the **Create an issue in Jira** dialog:



Note: The fields that are displayed might depend on the fields configured in the Jira project.

- a. Select the issue type that you want to create.
- b. Enter text for the title of the Jira issue. The name of the test is displayed as the default text in the **Summary** field.

You can view the details of the test run that is displayed in the **Description** field. These details are appended to the Jira issue under the **Description** field, after you create the issue.

- c. Select the existing labels in your Jira project or enter a new name for the label in the **Label** field, if required.



Note:

If the **Label** field is set to "Required" in your Jira project, you must select an existing label or create a new label by providing a name in the **Label** field. If you don't select an existing label or create a new label, an error message is displayed that indicates the **Label** field is a required field.

- d. Select the test reports or test logs of the test run that you want to attach to the Jira issue.



Note: You can select any report, test log, or all the reports and the test log as attachments.

- e. Click **Create**.

The issue is created in the project on the Jira server and the link to the issue is displayed in the **Results** panel of the test on Test Hub.



Note: You can create multiple issues for the same test result, if you want.

Results

You created issues in the Jira project for the test results from Test Hub.

The Jira issues are created in the Jira project and the links to the issues are displayed in the **Details** panel of the test result on the **Results** page.

What to do next

You can perform any of the following tasks:

- View details of the Jira issues that you created in Jira for the test. See [Viewing details of the Jira issues on page 902](#).
- Modify the configuration that you set for Jira on Test Hub. See [Managing the Jira configuration on page 903](#).

Related information

[Specifying Jira fields](#)

Viewing details of the Jira issues

After you create Jira issues in HCL DevOps Test Hub (Test Hub), you can view and track the details of the Jira issues in Jira.

Before you begin

You must have completed the following tasks:

- Completed a test run and the test results are available on the **Results** page.
- Created Jira issues against the test results. See [Creating Jira issues for the test results on page 895](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Analyze > Results**.

The **Results** page is displayed and all the test assets that were run earlier are displayed in tabs, as listed in the following table:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.



Note: You can click the relevant tab to view the tests that were run earlier.

4. Click the test that is listed under the **Test** column.
5. Click the **Report** tab in the **Results** panel, if it is not already displayed.
6. Click the Jira issue link under **Defect tracking**.

The Jira log-in page is displayed, if you are not already logged into Jira.

7. Enter your username and password.
8. Click **Login**.

The issue details are displayed in Jira.



Note: The **Description** field of the Jira issue in Azure DevOps provides additional details of the test that ran in Test Hub such as the name of the test, the person who ran the test, the verdict of the test result, the date and time of the test run, the links to the test report that is generated in Test Hub, and the link to the query that is created. You can also view the reports that you attached as a compressed file under the **Attachments** field in the Jira issue.

Results

You have viewed the details in Jira of the Jira issue that you created in Test Hub.

Managing the Jira configuration

You can manage the Jira configuration that you set in HCL DevOps Test Hub (Test Hub) when you want to modify the project configured in Jira or when you want to delete the existing Jira configuration.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned the role of a *Project Owner* in the project. See [Managing access to the server projects on page 1021](#).
- Ensured that you obtained the access and authentication details to Jira from the Jira administrator.

1. Log in to Test Hub.


The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Integration** in the navigation pane.

The **Integration** page is displayed.

4. Click the **CHANGE MANAGEMENT SYSTEM** tab.
5. Click the **Open action menu** icon  for the Jira server that you want to manage.

The **Configure** and **Delete** options are displayed in the list.

6. Perform any of the following actions when you want to modify or delete the Jira server configuration:
 - To modify the project in Jira that was set previously in the Jira configuration. Go to [Step 7 on page 904](#).
 - To delete the Jira configuration. Go to [Step 8 on page 904](#).
7. Perform the following steps to modify the project in Jira:
 - a. Select **Configure** as the option from the list.
The **Select project in Jira** dialog is displayed.
 - b. Select the project that you want from the list displayed in the **Project** field.
 - c. Click **Update** to save the updated project in the Jira server configuration.

8. Perform the following steps to delete the existing Jira server configuration:

- a. Select **Delete** as the option from the list.

The **Delete Jira configuration** dialog is displayed.

- b. Click **Delete**.

The Jira configuration is deleted and a confirmation message is displayed.



Note: The links to the issues that you created for the tests that ran are not deleted from the results panel in Test Hub even after you delete the Jira configuration.

Results

You achieved the following results:

- Updated the project in Jira and saved the Jira configuration.
- Deleted the Jira configuration.

Configuration of Azure DevOps as a change management system

You can use Azure DevOps as a change management system in HCL DevOps Test Hub (Test Hub) to create work items in the Azure DevOps project for the tests that ran in Test Hub. Links to the Azure DevOps work items are displayed on the **Results** page in Test Hub, and you can then view and track the work items in the Azure DevOps server.

Prerequisites

Before you configure Azure DevOps as a change management system in Test Hub, you must have completed the following tasks:

- Ensured that you obtained the access and authentication details to Azure DevOps from the Azure DevOps administrator.
- Ensured that you are assigned the role of a *Project Owner* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).

Supported installations

The following types of Azure DevOps installations are supported that can be configured as a change management system on Test Hub:

- On-premise Azure DevOps installations.
- On-cloud Azure DevOps installations.

Task flow

You can follow the tasks that are listed in the following table when you want to use Azure DevOps as a change management system in Test Hub. The table also provides you the links to the information about the tasks.

Task	Go to...
Configure Azure DevOps as a change management system.	Configuring Azure DevOps as a change management system on page 905
Create Azure DevOps work items for the test asset.	Creating Azure DevOps work items on page 907
Track and view the details of the created Azure DevOps work items.	Viewing details of the Azure DevOps work items on page 911

You can edit the saved configuration, or you can delete the configuration and create a new configuration. For more information, see [Managing the Azure DevOps configuration on page 913](#).

Configuring Azure DevOps as a change management system

You must configure Azure DevOps as a change management system in your project in HCL DevOps Test Hub (Test Hub) before you can create work items in the Azure DevOps project for the tests that ran.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Ensured that you obtained the access and authentication details to Azure DevOps from the Azure DevOps administrator.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Integration** in the navigation pane.

The **Integration** page is displayed.

4. Click the **CHANGE MANAGEMENT SYSTEM** tab.

5. Click **Add New**, and then select the **Azure DevOps** option from the drop-down list.

The **Configure Azure DevOps** dialog is displayed.

6. You can perform the following actions:

- a. Enter the URL of **Azure DevOps**.



Note: You must enter the URL of the Azure DevOps in one of the following formats:

- `https://dev.azure.com/{Organization}`
- `https://{Organization}.visualstudio.com`
- `https://{Hostname}:{Port}/{Collection}`

The URL contains information such as the server name (hostname, port, or domain name) and organization name.

- b. Click **Next**, and then paste the copied personal access token.



Note: If you do not have your personal access token, you can click **Generate** to generate your personal access token to authorize full access or access to the work items.

- c. Click **Next**, and then select an **Azure DevOps** project from the drop-down list.

- d. Click **Add**.

The configuration is added and saved to the **CHANGE MANAGEMENT SYSTEM** tab.

Results

You configured Azure DevOps as a change management system in Test Hub.

What to do next

You can create work items for the test results that are available in your project in Test Hub. See [Creating Azure DevOps work items on page 907](#).

Creating Azure DevOps work items

You can create Azure DevOps work items for the tests that ran in HCL DevOps Test Hub (Test Hub). The issues are created in the project in Azure DevOps that you configured in Test Hub.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Configured Azure DevOps as a change management system in Test Hub for your project. See [Configuring Azure DevOps as a change management system on page 905](#).
- Extended the CA list. See [Extending the trusted CA list on Linux on page 159](#) or [Extending the trusted CA list on Windows systems on page 159](#).
- Configured a test run. See [Test run configurations on page 446](#).
- Viewed the results of the completed test runs on the **Results** page. See [Test results on page 788](#).

About this task

When you want to create a Azure DevOps work item on Test Hub, the fields that display in the **Create a work item** dialog depends on the configuration done by the Azure DevOps administrator.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.


The **Overview** page of the project is displayed.

3. Click **Analyze > Results**.

The **Results** page is displayed and all the test assets that were run earlier are displayed in tabs, as listed in the following table:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.


Tab	Description
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.































 **Note:** You can click the relevant tab to view the tests that were run earlier.


4. Perform any of the following actions to identify the test asset:










- Scroll through the list.

 **Tip:** You can hover over the icon in the **Type** column to know the type of the test asset.

 **Note:** You can also identify the type of the asset from the icon that represents the test type as shown in the following table:

Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			





Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test by entering any text that is contained in the test asset name in the **Search** field.
- Select the time interval within which the test was run.

For example, you can select the *Last 24 hours* option to view the tests that were run in the previous 24 hours.

- Select the range of dates within which the test was run.


You can click the calendar icon  to select the date in the first image as the start date, and then select the date in the second image as the end date.

 **Note:** You can select the range of dates only when you select the time interval as *Custom*.

- Click the **Filter** icon  in the **SUITES** or **TESTS** tab, and then select the test type.

All the tests based on the selected test type are displayed.

For example, you can select **API Suite** in the **SUITES** tab to display all the API suites or you can select **Functional Test** in the **TESTS** tab to display all the functional tests.

- Click the **Filter** icon  in the **ADVANCED** tab, and then perform the following actions to create a filter query:
 - Click **New filter**.
 - Enter a name for the filter.
 - Select an operator, and then select **Add rule** to add a rule or **Add group** to add a group of rules.
 - Select or enter the relevant parameters, and then either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- ID
- Type

- Test
 - Version
 - Result
 - Started on
 - Finish on
 - Duration
 - Status
 - Verdict
 - Labels
 - Locked
 - Repository
 - Branch
 - Text
- e. Click **Save**.

The filter query is saved and all the tests that match the filter criteria are displayed.

- Perform the following actions in the **ADVANCED** tab to retrieve and apply a previously saved filter query:

- a. Click the **Filter** icon  .


All the filter queries that you created and saved earlier are displayed.


- b. Click the filter query that you want to apply.

All the tests that match the filter criteria are displayed.

You have identified the test that you want to use.

5. Perform the following actions on the test that was run to create a work item:

- a. Click the **Azure: Create work item** icon  that is under the **Actions** column in the row of the test.

Alternatively, you can click the test under the **Test** column to view the results panel, and then click the **Azure: Create work item** icon  that is displayed in the panel.

The **Create a work item** dialog is displayed.

- b. Perform the actions indicated in the following table for each of the fields:

Field	Description	Action
Project	The name of the Azure DevOps project.	Select the Azure DevOps project from the drop-down list.
Work item type	The type of work item.	Select the work item you want to assign for the test asset.

Field	Description	Action
Title	A title that identifies the work item.	Specify a title for the work item.
Description	A detailed description of the work item.	Enter a description of the work item.



Note: The other fields selected as **Required** fields on the **Azure DevOps** project are displayed in the **Create a work item** dialog.

- c. Enter the information for all the required fields, and then click **Create**.



Note: You can create multiple work items for the same test result, if required.

Results

You created work items in the Azure DevOps project for the test results from Test Hub.

The work items are created in the Azure DevOps project and the links to the work items are displayed in the **Details** panel of the test result on the **Results** page.

What to do next

You can perform any of the following tasks:

- View details of the Azure DevOps work items that you created in Azure DevOps for the test. See [Viewing details of the Azure DevOps work items on page 911](#).
- Modify the configuration that you set for Azure DevOps on Test Hub. See [Managing the Azure DevOps configuration on page 913](#).

Viewing details of the Azure DevOps work items

After you create a work item in HCL DevOps Test Hub (Test Hub), you can view and track the details of the Azure DevOps work item in Azure DevOps.

Before you begin

You must have completed the following tasks:

- Completed a test run and the test results are available on the **Results** page.
- Created Azure DevOps work item against the test results. See [Creating Azure DevOps work items on page 907](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

- Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

- Click **Analyze > Results**.

The **Results** page is displayed and all the test assets that were run earlier are displayed in tabs, as listed in the following table:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.



Note: You can click the relevant tab to view the tests that were run earlier.

- Click the test that is listed under the **Test** column.
- Click the **Report** tab in the **Results** panel, if it is not already displayed.
- Click the Azure DevOps work item link under **Defect tracking**.

The Azure DevOps log-in page is displayed.

- Enter your username and password.
- Click **Login**.

The work item details are displayed in Azure DevOps.



Note: The **Description** field of the work item in Azure DevOps provides additional details of the test that ran in Test Hub such as the name of the test, the person who ran the test, the verdict of the test result, the date and time of the test run, the links to the test report that is generated in Test Hub, and the link to the query that is created.

Results


You have viewed the details in Azure DevOps of the Azure DevOps work item that you created in Test Hub.

Managing the Azure DevOps configuration

You can edit or modify the Azure DevOps configuration that you added and saved to the Change management system tab on HCL DevOps Test Hub (Test Hub). If you do not want to use the existing configuration and want to set up a new Change management system configuration, then you can delete the configuration.

Before you begin

You must have completed the following task:

- Been assigned a project *Owner* role.
1. Open the project that you created on Test Hub.
 2. Go to **Manage**, and then click **Integration** in the navigation pane.
The **Integration** page is displayed.
 3. Click the **CHANGE MANAGEMENT SYSTEM** tab.
 4. Click **Open action menu** icon .

The **Edit** and **Delete** options are displayed in the drop-down list.

5. Perform any of the following actions:
 - To edit the Azure DevOps configuration. Go to [Step 6 on page 913](#).
 - To delete the Azure DevOps configuration. Go to [Step 7 on page 913](#).
6. Follow the step to edit the Azure DevOps configuration:
 - a. Select **Edit**.
The **Configure Azure DevOps** dialog is displayed.
 - b. Edit the URL by performing the following steps:
 - i. Enter the URL of **Azure DevOps**.
 - ii. Click **Next**.
 - iii. Paste the copied personal access token or click **Generate** to generate your personal access token.
 - iv. Click **Add**.

The URL is updated and saved.
 - c. Edit the project by performing the following steps:
 - i. Select an **Azure DevOps** project from the drop-down list.
 - ii. Click **Add**.

The project is updated and saved.
7. Follow these steps to delete the Azure DevOps configuration:
 - a. Select **Delete**.
A confirmation dialog is displayed.

b. Select **Delete**.

The Azure DevOps configuration is deleted.

Results

You achieved the following results:

- Edited the Azure DevOps configuration and saved the configuration.
- Deleted the Azure DevOps configuration.

Configuration of Rational® ClearQuest® or HCL® Compass as a change management system

You can use Rational® ClearQuest® or HCL® Compass as a change management system to create work items after you run tests on HCL DevOps Test Hub (Test Hub). You can then view and track the work items in Rational® ClearQuest® or HCL® Compass.

Prerequisites

Before you configure Rational® ClearQuest® or HCL® Compass on Test Hub, you must have completed the following tasks:

- Ensured that you obtained the access details and authentication details of Rational® ClearQuest® or Compass from the Rational® ClearQuest® or HCL® Compass administrator.
- Ensured that you are assigned the role of a *Project Owner* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).

Task flow

You can follow the tasks that are listed in the following table when you want to use Rational® ClearQuest® or HCL® Compass as a change management system on Test Hub. The table also provides links to the information about the tasks.

	Task	Go to...
1	Configure Rational® ClearQuest® or HCL® Compass as a change management system.	Configuring Rational ClearQuest or HCL Compass as a change management system on page 915
2	Create work items in Rational® ClearQuest® or HCL® Compass for the tests that ran on Test Hub.	Creating work items on page 916
3	View and track the details of the created work items.	Viewing details of the work items on page 921

You can manage the Rational® ClearQuest® or HCL® Compass settings that you configured in Test Hub. You can edit the saved configuration or you can delete the configuration. For more information, see [Managing the Rational ClearQuest or HCL Compass configuration on page 923](#).

Configuring Rational® ClearQuest® or HCL® Compass as a change management system

You can configure Rational® ClearQuest® or HCL® Compass as a change management system in your project to create work items from Test Hub in Rational® ClearQuest® or HCL® Compass.

Before you begin

You must have completed the following tasks:

- Read about the prerequisite tasks in [Prerequisites on page 914](#).
- Ensured that you are assigned the role of a *Project Owner* in the project. See [Managing access to the server projects on page 1021](#).
- Ensured that you obtained the access details and authentication details of Rational® ClearQuest® or Compass from the Rational® ClearQuest® or HCL® Compass administrator.
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Integration** in the navigation pane.

The **Integration** page is displayed.

4. Click the **CHANGE MANAGEMENT SYSTEM** tab.

5. Click **Add New**, and then select the **DevOps Plan** option from the list.

The **Configure DevOps Plan application** dialog is displayed.

6. Configure the Rational® ClearQuest® or HCL® Compass application by performing the following steps:

- a. Enter the URL of the Rational® ClearQuest® or HCL® Compass server.



Note: You can enter the URL of Rational® ClearQuest® or HCL® Compass in any of the following formats:

- `https://IP_address:port/`
- `https://hostname/`



Where	Is
<i>IP_address</i>	The IP address of the server on which Rational® ClearQuest® or HCL® Compass is installed.
<i>port</i>	The port that is used to communicate with Rational® ClearQuest® or HCL® Compass.
<i>hostname</i>	The hostname of the computer on which Rational® ClearQuest® or HCL® Compass is installed.

- b. Click **Next**, and then select the application from the **Application** list.
- c. Click **Add**.

The configuration is added and saved to the **CHANGE MANAGEMENT SYSTEM** tab. You can view the information in the **Configuration status** column to verify if the connection is established.

Results

You have configured Rational® ClearQuest® or HCL® Compass as a change management system in your project on Test Hub.

What to do next

You can perform any of the following tasks:

- Create work items in Rational® ClearQuest® or HCL® Compass for the tests that you ran on Test Hub. See [Creating work items on page 916](#).
- Modify the configuration that you set for Rational® ClearQuest® or HCL® Compass on Test Hub. See [Managing the Rational ClearQuest or HCL Compass configuration on page 923](#).

Creating work items

You can create work items in Rational® ClearQuest® or HCL® Compass for the tests that ran on HCL DevOps Test Hub (Test Hub). For example, you might want to create a work item in Rational® ClearQuest® or HCL® Compass when the test results differ from the expected results.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Configured Rational® ClearQuest® or HCL® Compass as a change management system in Test Hub. See [Configuring Rational ClearQuest or HCL Compass as a change management system on page 915](#).

- Configured applications in Rational® ClearQuest® or HCL® Compass.
- Configured a test run. See [Test run configurations on page 446](#).
- Viewed the results of the completed test runs on the **Results** page. See [Test results on page 788](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Analyze > Results**.

The **Results** page is displayed and all the test assets that were run earlier are displayed in tabs, as listed in the following table:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.



Note: You can click the relevant tab to view the tests that were run earlier.

4. Perform any of the following actions to identify the test asset:




- Scroll through the list.








































Tip: You can hover over the icon in the **Type** column to know the type of the test asset.



Note: You can also identify the type of the asset from the icon that represents the test type as shown in the following table:

Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the ADVANCED tab
	API test			




Icon	Represents	Listed in the SUITES tab	Listed in the TESTS tab	Listed in the AD-VANCED tab
	Functional test			
	Performance test			
	AFT Suite			
	API Suite			
	Compound Test			
	HCL® AppScan CodeSweep			
	JMeter Test			
	JUnit Test			
	Postman test			
	Rate Schedule			
	Selenium test			
	VU Schedule			

- Search for the test by entering any text that is contained in the test asset name in the **Search** field.
- Select the time interval within which the test was run.


For example, you can select the *Last 24 hours* option to view the tests that were run in the previous 24 hours.

- Select the range of dates within which the test was run.

You can click the calendar icon  to select the date in the first image as the start date, and then select the date in the second image as the end date.




Note: You can select the range of dates only when you select the time interval as *Custom*.

- Click the **Filter** icon  in the **SUITES** or **TESTS** tab, and then select the test type.

All the tests based on the selected test type are displayed.


For example, you can select **API Suite** in the **SUITES** tab to display all the API suites or you can select **Functional Test** in the **TESTS** tab to display all the functional tests.

- Click the **Filter** icon  in the **ADVANCED** tab, and then perform the following actions to create a filter query:
 - a. Click **New filter**.
 - b. Enter a name for the filter.
 - c. Select an operator, and then select **Add rule** to add a rule or **Add group** to add a group of rules.
 - d. Select or enter the relevant parameters, and then either select or enter the condition and the criteria for the condition.

You can select a parameter from the following list:

- ID
 - Type
 - Test
 - Version
 - Result
 - Started on
 - Finish on
 - Duration
 - Status
 - Verdict
 - Labels
 - Locked
 - Repository
 - Branch
 - Text
- e. Click **Save**.

The filter query is saved and all the tests that match the filter criteria are displayed.

- Perform the following actions in the **ADVANCED** tab to retrieve and apply a previously saved filter query:
 - a. Click the **Filter** icon .


All the filter queries that you created and saved earlier are displayed.


- b. Click the filter query that you want to apply.

All the tests that match the filter criteria are displayed.

You have identified the test that you want to use.

5. Perform the following actions on the identified test to create a work item in Rational® ClearQuest® or HCL® Compass:

- a. Click the **DevOps Plan: Create a work item** icon  that is under the **Actions** column in the row of the test.

Alternatively, you can click the test under the **Test** column to view the results panel, and then click the **DevOps Plan: Create a work item** icon  that is displayed in the panel.

The **Create a work item in DevOps Plan** dialog is displayed.

- b. Select the application from the list in the **Application** field.



Remember: The application that you select must be the application in Rational® ClearQuest® or HCL® Compass for which you ran the test.

- c. Enter the username in the **Username** field.



Note: The username is authorized to access the Rational® ClearQuest® or HCL® Compass server.

- d. Enter the password in the **Password** field to authenticate the username.



Note: After the credentials that you provided in the **Username** and **Password** fields are authorized by the Rational® ClearQuest® or HCL® Compass server, you need not enter these values for subsequent connections to create a work item.

- e. Click **Next**.

The dialog displays the URL of the Rational® ClearQuest® or HCL® Compass server and the **Application** that you selected.

- f. Select the work item type as *WorkItem*.

- g. Click **Next**.

- h. Select the project from the list in the **Project** field.

- i. Select the type of the work item from the list in the **Type** field.

- j. Enter text that is to be used as the title of the work item in the **Title** field. The text entered is displayed as the title in the work item in Rational® ClearQuest® or HCL® Compass.

You can view the details of the test run in the **Description** field, which is appended to the work item that is created for the test run in Rational® ClearQuest® or HCL® Compass.

- k. Click **Create**.

A work item is created in Rational® ClearQuest® or HCL® Compass and a link to the work item is created in the **Report** panel of the **Results** page.



Notes:

- You cannot delete the work item that is created in Rational® ClearQuest® or HCL® Compass from Test Hub.
- You can create multiple work items for the same test result, if required.

Results

You have created a work item in Rational® ClearQuest® or HCL® Compass from the **Results** page for the test that ran on Test Hub.

What to do next

You can perform any of the following tasks:

- View details of the work items that you created in Rational® ClearQuest® or HCL® Compass for the test. See [Viewing details of the work items on page 921](#).
- Modify the configuration that you set for Rational® ClearQuest® or HCL® Compass on Test Hub. See [Managing the Rational ClearQuest or HCL Compass configuration on page 923](#).

Viewing details of the work items

After you create a work item in Rational® ClearQuest® or HCL® Compass from HCL DevOps Test Hub (Test Hub), you can view and track the details of the work item in Rational® ClearQuest® or HCL® Compass.

Before you begin

You must have completed the following tasks:

- Completed a test run and the test results are available on the **Results** page.
- Created a work item in Rational® ClearQuest® or HCL® Compass against the test results. See [Creating work items on page 916](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

- Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

- Click **Analyze > Results**.

The **Results** page is displayed and all the test assets that were run earlier are displayed in tabs, as listed in the following table:

Tab	Description
SUITES	Lists all suites, Compound Tests, JMeter tests, JUnit tests, Postman tests, Rate Schedules, Selenium tests, or VU Schedules that are in the selected branch of the repository.
TESTS	Lists all API tests, functional tests, or performance tests that are in the selected branch of the repository.
ADVANCED	Lists all assets that are displayed when custom filters are applied for assets in the selected branch of the repository.



Note: You can click the relevant tab to view the tests that were run earlier.

- Click the test that is listed under the **Test** column.

The **Results** panel for the test is displayed.

- Click the **Report** tab in the **Results** panel, if it is not already displayed.
- Click the work item link under **Defect tracking**.

The Rational® ClearQuest® or HCL® Compass log-in page is displayed.

- Enter your username and password.
- Click **Sign In**.

The work item details are displayed in Rational® ClearQuest® or HCL® Compass.



Note: The **Description** field of the work item provides additional details of the test that ran in Test Hub such as the name of the test, the person who ran the test, the verdict of the test result, the date and time of the test run, the links to the test report that is generated in Test Hub, and the link to the query that is created.

Results

You have viewed the details in Rational® ClearQuest® or HCL® Compass of the work item that you created in Test Hub.

Managing the Rational® ClearQuest® or HCL® Compass configuration

You can manage the Rational® ClearQuest® or HCL® Compass configuration that you set in HCL DevOps Test Hub (Test Hub) when you want to edit the configuration or when you want to delete the existing configuration.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned the role of a *Project Owner* in the project. See [Managing access to the server projects on page 1021](#).
- Ensured that you obtained the access details and authentication details of Rational® ClearQuest® or Compass from the Rational® ClearQuest® or HCL® Compass administrator.

1. Log in to Test Hub.

The team space that contains your project is displayed.


2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Integration** in the navigation pane.

The **Integration** page is displayed.

4. Click the **CHANGE MANAGEMENT SYSTEM** tab.

5. Click the **Open action menu** icon  for the Rational® ClearQuest® or HCL® Compass server that you want to manage.

The **Edit** and **Delete** options are displayed in the list.

6. Perform any of the following actions when you want to edit or delete the Rational® ClearQuest® or HCL® Compass configuration:

- To edit the Rational® ClearQuest® or HCL® Compass configuration. Go to [Step 7 on page 923](#).
- To delete the Rational® ClearQuest® or HCL® Compass configuration. Go to [Step 8 on page 924](#).

7. Perform the following steps to edit the Rational® ClearQuest® or HCL® Compass application that is configured:

- a. Select **Edit** as the option in the list.

The **Modify DevOps Plan configuration** dialog is displayed.

- b. Edit the URL or change it to the one that you want to use in the **URL** field, and then select the application that you want to use from the list.

- c. Click **Update**.

The Rational® ClearQuest® or HCL® Compass configuration is updated and a confirmation message is displayed.

- d. Go to [Results on page 924](#).

8. Perform the following steps to delete the existing Rational® ClearQuest® or HCL® Compass configuration:

a. Select **Delete** as the option in the list.

The **Delete DevOps Plan configuration** dialog is displayed.

b. Click **Delete**.

The Rational® ClearQuest® or HCL® Compass configuration is deleted and a confirmation message is displayed.

c. Go to [Results on page 924](#).



Note: The links to the records that you created for the tests that you ran are not deleted from the results panel in Test Hub even after you delete the Rational® ClearQuest® or HCL® Compass configuration.

Results

You achieved the following results:

- Edited and saved the Rational® ClearQuest® or HCL® Compass configuration.
- Deleted the Rational® ClearQuest® or HCL® Compass configuration.

Integrations with other applications

You can integrate certain applications with HCL DevOps Test Hub (Test Hub) to run tests, view the test results, and create defects.

Integration plugin compatibility matrix

You can find information about the versions of the integration plugin that are compatible with HCL DevOps Test Hub (Test Hub).

The following table lists the versions of the integration plugin that are required to integrate Test Hub with DevOps Deploy, IBM® Engineering Test Management, and Jenkins.



Note: You must download the required version of the integration plugin from [HCL® License & Delivery portal](#) based on the existing version of Test Hub. You can then integrate Test Hub with HCL DevOps Deploy, IBM® Engineering Test Management, and Jenkins.

Test Hub	Engineering Test Management plugin	Jenkins plugin	HCL DevOps Deploy plugin
11.0.1	HCL_DT_HUB_1101_ETM	HCL_DT_HUB_1101_Jenkins	HCL_DT_HUB_1101_- Launch HCL_DT_HUB_1101_UCD
11.0.0	HCL_DT_HUB_110_ETM	HCL_DT_HUB_110_Jenkins	HCL_DT_HUB_110_Launch
10.5.4	HCL_OT_SERVER_1054_- ETM	HCL_OT_SERVER_1054_- Jenkins	HCL_OT_SERVER_1054_- Launch HCL_OT_SERVER_1054_- UCD
10.5.3	HCL_OT_SERVER_1053_- ETM	HCL_OT_SERVER_1053_- Jenkins	HCL_OT_SERVER_1053_- Launch HCL_OT_SERVER_1053_- UCD
10.5.2	HCL_OT_SERVER_1053_- ETM	HCL_OT_SERVER_1052_- Jenkins	HCL_OT_SERVER_1052_- Launch HCL_OT_SERVER_1052_- UCD
10.5.1	HCL_OT_SERVER_1051_- ETM	HCL_OT_SERVER_1051_- Jenkins	HCL_OT_SERVER_1051_- Launch HCL_OT_SERVER_1051_- UCD
10.5.0	HCL_OT_SERVER_1050_- ETM	HOT-SERVER-Jenkins-6.3	HOT-SERVER-UCD-3.3

Test Hub	Engineering Test Management plugin	Jenkins plugin	HCL DevOps Deploy plugin
10.2.3	HCL_OT_SERVER_1023_-ETM	HOT-SERVER-Jenkins-6.1	HOT-SERVER-UCD-3.2
10.2.2	HCL_OT_SERVER_1022_-ETM	HOT-SERVER-Jenkins-6.0	HOT-SERVER-UCD-3.0
10.2.1	HCL_OT_SERVER_1021_-ETM	HOT-SERVER-Jenkins-5.0	HOT-SERVER-UCD-3.0
10.2.0	NA	HOT-SERVER-Jenkins-5.0	HOT-SERVER-UCD-3.0
10.1.3	NA	HOT-SERVER-Jenkins-4.0	HOT-SERVER-UCD-2.0
10.1.2	NA	HOT-SERVER-Jenkins-4.0	HOT-SERVER-UCD-2.0
10.1.1	NA	HOT-SERVER-Jenkins-3.0	HOT-SERVER-UCD-1.0
10.1.0	NA	HOT-SERVER-Jenkins-2.0	HOT-SERVER-UCD-1.0

Integration with Azure DevOps

When you use Azure DevOps for continuous integration and continuous development of your application, you can run tests created for your application and available in a project on HCL DevOps Test Hub (Test Hub), in Azure DevOps pipelines by using the HCL DevOps Test Hub extension.

Overview

You can use the HCL DevOps Test Hub extension to integrate Test Hub with Azure DevOps.

The HCL DevOps Test Hub extension enables you to select any type of test available for your project in Test Hub that you can add to your task for the job in the Azure DevOps pipelines.

You can now follow the tasks listed in the task flow table to integrate Test Hub with Azure DevOps. See [Task flow for integrating Azure DevOps on page 927](#).

Task flow for integrating DevOps Test Hub with Azure DevOps

The table shows the task flow for integrating HCL DevOps Test Hub (Test Hub) with Azure DevOps by using the HCL DevOps Test Hub extension. You must perform these tasks in sequence as listed in the following table. The table also provides you the links to the information about the tasks.

S.N.	Tasks	More information
1	Install Test Hub.	Installation of DevOps Test Hub on page 93
2	Create tests in the desktop clients for the application that you want to test.	<ul style="list-style-type: none"> • DevOps Test Performance documentation • DevOps Test UI documentation • DevOps Test Integrations and APIs documentation
3	Create a project on Test Hub and add the test assets that are created in the desktop clients to your project.	Managing DevOps Test Hub projects on page 1010
4	Create an organization and a project in Azure DevOps for running jobs in Azure DevOps pipelines.	Creating an organization
5	Access the Visual Studio Marketplace portal and search for the latest version of the HCL DevOps Test Hub extension.	Visual Studio Marketplace
6	Install the latest version of the HCL DevOps Test Hub extension.	Installing the HCL DevOps Test Hub extension on page 928
7	Extend the trust certificate if you have used an internal CA certificate.	Certificate authority: Importing and extending lists on page 156
8	Run tests in an Azure DevOps pipeline.	Running tests in an Azure DevOps Pipeline on page 929

Related information

[Integration with Azure DevOps on page 926](#)

Installing the HCL DevOps Test Hub extension

You must install the latest version of the HCL DevOps Test Hub extension in your Azure DevOps organization before you can use the extension for running your application tests in an Azure DevOps pipeline.

Before you begin

You must have access to the **Visual Studio Marketplace** portal to install the latest version of the HCL DevOps Test Hub extension.

About this task

After you install the latest version of the HCL DevOps Test Hub extension from the **Visual Studio Marketplace** portal in your Azure DevOps organization, you can select the tests that you want to run for your application in an Azure DevOps pipeline by using the HCL DevOps Test Hub extension.

1. Log in to the **Visual Studio Marketplace** portal, if you are not already logged in.
2. Click the **Azure DevOps** tab.
3. Search for the **HCL DevOps Test Hub extension**.
4. Click **HCL DevOps Test Hub extension**.
5. Click **Get it free**.

Result

The **Visual Studio Marketplace** portal for the HCL DevOps Test Hub extension is displayed.

6. Select the organization where you want to run your test from the **Select an Azure DevOps Organization** list.
7. Click **Install**.

Result

The installation is completed.

8. Click **Proceed to organization**.

Result

The **Organization** page in Azure DevOps is displayed.

9. Click **Organization settings > Extensions**.

Result

The HCL DevOps Test Hub extension is displayed as an installed extension.

Results

You have installed the HCL DevOps Test Hub extension in your Azure DevOps organization.

What to do next

You can run tests that are available in Test Hub as a job in an Azure DevOps pipeline. See [Running tests in an Azure DevOps Pipeline on page 929](#).

Running DevOps Test Hub tests in an Azure DevOps Pipeline

After you install the HCL DevOps Test Hub extension in your organization, you can run tests that are available in HCL DevOps Test Hub (Test Hub) as a job in Azure DevOps pipelines.

Before you begin

You must have completed the following tasks:

- Added the tests that you created in the desktop clients for your application to the project on Test Hub.
- Installed the latest version of the HCL DevOps Test Hub extension in your organization. See [Installing the HCL DevOps Test Hub extension on page 928](#).
- Installed an agent in your pipeline. See [Azure Pipelines agents](#).
- Extended the CA list. See [Extending the trusted CA list on Linux on page 159](#) or [Extending the trusted CA list on Windows systems on page 159](#).
- Generated an offline user token from Test Hub. See [Generating an offline token on page 994](#).
- Created test cases under test plans if you want to view results of the test runs on the Test Plans dashboard.
- Created encrypted secrets or environment variables for your personal access token (PAT), if you want to use an encrypted PAT.

About this task

After you add the HCL DevOps Test Hub extension in your Azure DevOps organization, you can use an existing pipeline or create a new one to add Test Hub test tasks. You can install an agent or use the one that you installed in your default agent pool. You can add Test Hub tests as tasks to your agent job, configure the task, and then run the task in the Azure DevOps pipeline.

If you have created test cases under test plans in a project on Azure DevOps, you must provide the details of the Azure DevOps URL, test plan, test case, and your PAT while you configure the test job in a pipeline so that you can view the results of the test run on your Test Plan dashboard.


1. Open your **Organization** page in Azure DevOps and perform the following steps:
 - a. Click the project you want to use.
 - b. Initialize the repository by performing the following steps:
 - i. Click **Repos** from the left pane.
 - ii. Click **Initialize** from the **Initialize with a README or gitignore** section.



Note: Select the **Add a README** checkbox if it is not selected.

- c. Click **Pipelines** from the left pane.
- d. Click **Create Pipeline**.
- e. Click **Use the classic editor** to create a pipeline without YAML.

- f. Verify the project, repository, and branch for manual and scheduled builds, and then click **Continue**.
 - g. Click **Empty job**.
 2. Select **Pipeline** and complete the following steps:
 - a. Change the name for the build pipeline if required.
 - b. Select the **Agent pool** for your build pipeline.

You can use the agent from the default agent pool or use the one you have installed.
 - c. Select the **Agent Specification** for the agent if required.
 3. Add a task to the agent job by completing the following steps:
 - a. Click the **Add Task** icon  for the agent job.

Result

The **Add tasks** pane is displayed.
 - b. Search for the HCL tasks defined in the HCL DevOps Test Hub extension.

Result

The tasks that you can select are displayed.
 - c. Select the **HCL DevOps Test Hub Task**, and then click **Add** to add the task to the agent job.

Result

The selected task is added to the agent job and it is displayed with a warning that some settings require attention. You must configure the settings mentioned in [Step 4 on page 930](#).

You can also remove the tasks that are not required in your job. Select the tasks in the list that you want to remove. You can then right-click the tasks, and click **Remove selected task(s)** to remove them.
 4. Configure the settings by performing the following steps:
 - a. Select the task version from the list if required.
 - b. Follow the action for the task by referring to the following table:







Note: All mandatory fields are marked with an asterisk (*) in the UI.



Field	Description	Action
Display name	The default name of the task. The default name that is displayed is HCL DevOps Test Hub Task .	Enter a name for the task that helps you to identify the task by changing the default name.
HCL DevOps Test Hub Task service connection	The details of Test Hub to set up a connection with Azure DevOps.	<p>Select the service connection from the drop-down list.</p> <p>If you are selecting the Test Hub service connection for the first time, you must click New to add the following details to add Test Hub service connection, and then save the connection details:</p> <ul style="list-style-type: none"> ▪ Server URL - Enter the URL of Test Hub. The format of the URL is as follows: <code>https://hostname</code>. ▪ Offline Token - Enter the offline token that you generated from Test Hub. ▪ Service connection name - Enter a service connection name. ▪ Description (optional) - Enter the details of the service connection if required. ▪ Optionally, select the Grant access permission to all pipelines checkbox if required. <p>You can save the service connection details. The service connection is available for selection from the Test Hub service connection drop-down list.</p>






Note:

You can edit or delete the service connection that you added if required. Click **Manage**, and then select the service connection from the **Service connections** list to open the service connection. You can click **Edit** to edit the service connection details. If you want to delete an existing service connection details, click the **Vertical**


Field	Description	Action
		 ellipsis icon, and then click Delete to delete the service connection details.
		<p>Optionally, you can also create, edit, and delete a service connection from your Azure DevOps project dashboard. Open your project, click Project settings > Service connections, and then perform any of the following tasks:</p> <ul style="list-style-type: none"> ▪ Create a new service connection. ▪ Edit or delete an existing service connection that you already added.
Team Space Name	The name of the team space on Test Hub that contains your project with the test assets.	<p>Enter the name of the team space that contains the project.</p>  Note: The license on the team space must be configured and you must be a member of that team space.
Project	The name of the project containing the test assets in your team space on Test Hub.	<p>Enter the name of the project from the available names in the team space.</p>  Note: <p> Note: The Project drop-down list displays the projects where you are an <i>Owner</i> or a <i>member</i> of Test Hub. You must be an <i>Owner</i> or a <i>Tester</i> of the project that is available in the team spaces to run the tests from the Jenkins server.</p> <p>Azure DevOps pipeline.</p>
Branch	The name of the branch of the repository in your project on Test Hub that contains the test assets.	Enter the branch where the test assets are stored.
Repository Link	The URL or link to the repository in your project that contains the test assets.	Enter the repository path that is configured for the project that contains the test to run.

Field	Description	Action
File Path	The path of the test in the repository of your project.	Enter the path of the test in Test Hub that you want to run.
API Test Environment	The environment defined in the API Suite or test to use in the test run.	Enter the name of the environment that is configured in the test asset for the test created in Test Integrations and APIs.  Note: This field is applicable only if you are running an API suite test.
Variables	The variable name and its value.	Enter the name of the variable and its value if your test requires variables during the test run time. You must enter the variables in the following format: <code>name_of_the_variable=value_of_the_varibale</code> You can add multiple variables to the test run separated by a semicolon. For example, <code>varname1=value1;varname2=value2</code>
Datasets	The name with path to the dataset that must be used during the test run.	Enter the path to the dataset if you want to replace the values of the dataset during a test run. You must ensure that both original and new datasets are in the same workspace and have the same column names. When you enter a value for the Datasets field, you must also include the path to the dataset. The path must be in the following format: <code>/project_name/ds_path/original_datasets.csv:/project_name/ds_path/new_ds.csv</code>  Note: You can override multiple datasets that are saved in a different project by adding multiple paths to the dataset separated by a semicolon.
Labels	The option to add labels to the test results when the test run is complete.	Enter the labels to add to test results when the test run is complete. You can add multiple labels to a test result separated by a comma.

Field	Description	Action
		<p>For example, <i>label1</i>, <i>label2</i></p> <p>After the test run is complete, then the values provided in the Labels field are displayed on the Results page of Test Hub.</p>
Classification Name	<p>The name of the classification that contains the secrets to be used in certain API Suites or tests.</p>	<p>Enter the name of the classification that contains the secrets to be used in the API Suite or test.</p> <p> Note: This field is applicable only if you are running an API suite test.</p>
Azure DevOps Project URL	<p>The URL of the test project in the organization on the Azure DevOps server.</p>	<p>Enter the URL of the test project in the organization on the Azure DevOps server in the following format:</p> <p><code>https://<host>/<orgname>/<projectName></code></p> <p>You must use this option with the following options:</p> <ul style="list-style-type: none"> ▪ Azure DevOps PAT ▪ Test Plan Name ▪ Test Case Name <p> Note: You must enter the details for this option if you have created test cases under test plans in your Azure project and want to view the test results on the Test Plans Dashboard.</p>
Azure DevOps PAT	<p>The personal access token (PAT) of the Azure server where the test project that contains the test plan is hosted. You can also enter the variable for the encrypted PAT that you create in Azure DevOps.</p>	<p>Enter your personal access token.</p> <p>You must use this option with the following options:</p> <ul style="list-style-type: none"> ▪ Azure DevOps Project URL ▪ Test Plan Name ▪ Test Case Name

Field	Description	Action
	 Note: The token must have read and write access.	
Test Plan Name	The name of the test plan in your Azure project.	Enter the name of the test plan. You must use this option with the following options: <ul style="list-style-type: none"> ▪ Azure DevOps Project URL ▪ Azure DevOps PAT ▪ Test Case Name
Test Case Name	The name of the test case under the test plan in your Azure project.	Enter the name of the test case. You must use this option with the following options: <ul style="list-style-type: none"> ▪ Azure DevOps Project URL ▪ Azure DevOps PAT ▪ Test Plan Name
Export JUnit Report	The location to store the JUnit report of the test.	Enter the path to store the JUnit report that is generated for the test in the XML format. You must specify the complete path to the directory including the file name of the report. For example, <code>C:/Users/Documents/Reports/junit_test_report.xml</code>

- c. Expand **Control Options** and configure the settings for your task if required.
 - d. Expand **Output Variables** and configure the settings for your task if required.
5. Select from the following options:
- a. Click **Save** to save the configured settings for the task.

 **Note:** The task is not queued for a run.

You can save the task to a build pipeline and opt to run the build at a later time.

- b. Click **Save & queue** to save the configurations and queue the run in the pipeline.

Result

The **Run pipeline** dialog is displayed.

6. Complete the following steps:

- a. Enter a comment for the test in the **Save comment** field.
- b. Select the agent that you configured for the test from the **Agent pool** list.
- c. Select the agent specification from the **Agent Specification** list if required.
- d. Select the branch from the **Branch/tag** list.
- e. Add the variables and demands for the task run from the **Advanced Options** pane if required.
- f. Select the **Enable system diagnostics** checkbox for a detailed log view.
- g. Click **Save and run**.

Result

The `pipeline summary` page displays the progress of the job run.

Results

You have run the tests for the application that you are testing in the Azure DevOps pipeline.

What to do next

You can perform any of the following actions in Azure DevOps:

- View the test results on the **Task** page by clicking the task.

The **Reports information** section on the **Task** page displays the names of the report along with its corresponding URLs. The report URLs are the Test Hub URLs where the reports are stored. You can access the report URLs to view the test run information at any point of time.

- View the task logs from the **pipeline summary** page for the job you ran.
- View the test results on the Test Plans Dashboard if you created test cases under test plans in your Azure project.

You can perform any of the following actions in Test Hub:

- View the test results of the test that you ran from Azure DevOps on the **Results** page.
- View the logs and reports of the test that you ran from Azure DevOps by clicking the result panel of the test asset or resource, and then click the log or the report you want to view.

Integration with HCL DevOps Deploy

You can integrate HCL DevOps Test Hub (Test Hub) with HCL DevOps Deploy (Deploy) to run test assets as a **Process** from Deploy.

When you use Deploy for automating the deployments of your applications to an environment during the development cycle, you can create test assets for your application either in Test Hub or desktop clients. After you configure the project in Test Hub, you can then run those tests assets from Deploy.

You must use the Test Hub Deploy plugin to integrate Test Hub with Deploy. Integrating Test Hub with Deploy automates the process of test runs. If you have many tests to run at regular intervals, you can use Deploy to initiate test runs automatically.

The following table lists the tasks that you must perform to run performance test assets from Deploy:

Tasks	Go to...
Create any or all the following types of test assets in desktop clients to test your application: <ul style="list-style-type: none"> • API test assets • Functional test assets • Performance test assets 	<ul style="list-style-type: none"> • DevOps Test Performance documentation • DevOps Test UI documentation • DevOps Test Integrations and APIs documentation
Create a project, and then configure the repository in Test Hub.	Working with projects in a team space on page 1010
Install Deploy with at least one agent.	HCL® Launch Documentation . System Requirements on page 83 for information about specific versions of Deploy requirements.
Install the DevOps Test Hub Deploy plugin on Deploy.	Installing the plugin on HCL DevOps Deploy on page 938 .
Create a Component in Deploy.	Creating a component in HCL DevOps Deploy on page 940 .
Create a Process for the component in Deploy.	Creating a process in HCL DevOps Deploy on page 940 .
Configure the Process that you created for the component in Deploy.	Configuring the process in HCL DevOps Deploy on page 942 .
Create a Resource in Deploy.	Creating a resource in HCL DevOps Deploy on page 946 .
Configure the Resource in Deploy.	Configuring the resource in HCL DevOps Deploy on page 946 .
Create an Application in Deploy.	Creating an application in HCL DevOps Deploy on page 948 .

Tasks	Go to...
Configure the Application that you use to run performance test assets in Deploy.	Configuring the application HCL DevOps Deploy on page 948.
Run the test assets from Deploy.	Running tests from HCL DevOps Deploy on page 950.

Installing the plugin on HCL DevOps Deploy

You must install the DevOps Test Hub Deploy plugin to run test assets from HCL DevOps Deploy (Deploy).

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Installed Deploy. For more information about the installation, refer to the [HCL Launch](#) documentation.
 - Been granted access to Deploy.
 - Downloaded the DevOps Test Hub Deploy plugin from the [HCL® License & Delivery portal](#).

For more information about specific versions of plug-in, see [Integration plugin compatibility matrix on page 924](#).

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Settings**.
3. Click **Automation Plugins** from the **Automation** pane.
4. Click **Load plugin**.
5. Click **Choose File** to locate and open the compressed DevOps Test Hub Deploy plugin file.



Remember: You must not extract the DevOps Test Hub Deploy plugin compressed file contents.

6. Click **Submit**.

Result

The installed plugin is displayed on the **Automation Plugins** tab.

Results

You have installed the DevOps Test Hub Deploy plugin on Deploy.

What to do next

You must create a **Component** in Deploy. See [Creating a component in HCL DevOps Deploy on page 940](#).

Using a custom trust store for HCL DevOps Deploy

You can use a custom trust store in the DevOps Test Hub Deploy plugin file to establish a trusted and secure connection between the HCL DevOps Deploy (Deploy) server and HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

You must have completed the following tasks:

- Saved the Certificate Authority (CA) assigned to Test Hub.



Note: You can contact the administrator of Test Hub if you do not have a copy of the CA.

- Installed Java on the computer from where you access the Test Hub UI.

About this task

If the SSL certificate assigned to Test Hub is signed by an internal Certified Authority (CA), then you must download and import the CA certificate to a custom trust store. You can then use the custom trust store in the Test Hub test process step to establish a trusted and secure connection between the Deploy server and Test Hub.



Note: If the internal CA certificate is already imported to the default trust store that is used by the Deploy server, you need not use a custom trust store.

1. Locate the default trust store file (`cacerts` file) from the JRE directory.
For example: `C:\Program Files\Java\jdk-18.0.1.1\lib\security`
2. Copy the `cacerts` file to a location of your choice on your computer.
3. Place the CA assigned to Test Hub in the same location where you saved the `cacerts` file.
4. Run the following command from the command-line interface or terminal to import the CA of Test Hub to custom trust store:

```
keytool -import -trustcacerts -file {path to the downloaded CA with the file extension} -alias {label for the certificate} -keystore {path to the trust store cacerts file}
```

For example,

```
keytool -import -trustcacerts -file C:\Users\Desktop\cert\ca-master-hcl.crt -alias "otsca" -keystore C:\Users\Desktop\cert\cacerts
```



Note: The default password of the trust store is *changeit*. It remains the same for the custom trust store. If you want to change the password, you can run the following command, and then enter the new password:



```
keytool -storepasswd -keystore {path to the trust store cacerts file}
```

For example, `keytool -storepasswd -keystore C:\Users\Desktop\cert\cacerts`

Results

You have successfully imported the downloaded CA to the custom trust store.

What to do next

You can then run the tests that are available in your Test Hub project from the Deploy server.

Creating a component in HCL DevOps Deploy

You must create a component to include artifacts and processes. The artifacts include runnable files, images, databases, configuration instructions. Whereas the processes define the activities that components can perform.

Before you begin

- You must be familiar with working with Deploy.
- You must have been granted access to Deploy.

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Components**, and then click **Create Component**.
3. Enter a name for the component in the **Name** field.
4. Enter the details in the other optional fields based on your requirement, and then click **Save**.

Result

The component that you created is displayed.

Results

You have created the component in Deploy.

What to do next

You must create a process for the component in Deploy. See [Creating a process in HCL DevOps Deploy on page 940](#).

Related information

[HCL DevOps Deploy Documentation](#)

Creating a process in HCL DevOps Deploy

You must create a process for the component to include step properties for the test that you want to run from HCL DevOps Deploy (Deploy).

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Been granted access to Deploy.
 - Created a component in Deploy. See [Creating a component in HCL DevOps Deploy on page 940](#).

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Components**.

Result

A list of components that are available in Deploy is displayed.

3. Select the component from the list for which you want to create a process.
4. Click the **Processes** tab, and then click **Create Process**.

Result

The **Create Process** dialog is displayed.

5. Enter a name for the process in the **Name** field.
6. Select **Operational (No Version Needed)** from **Process Type** drop-down list.
7. Verify the **Default Working Directory** field.

The **Default Working Directory** field defines the location that the agent uses to run the process. The default value is `${p:resource/work.dir}/${p:component.name}`.

Where `${p:resource/work.dir}` is the default working directory for the agent and `${p:component.name}` is the name of the component.

8. Click **Save**.

Result

The process that you created is listed in the **Processes** tab and the **Design** tab for the process is displayed.

Results

You have created the process for the component in Deploy.

What to do next

You must configure the process in Deploy. See [Configuring the process in HCL DevOps Deploy on page 942](#).

Related information

[HCL DevOps Deploy Documentation](#)

Configuring the process in HCL DevOps Deploy

You must configure the process that you created for the component to organize the steps in the process, specify the properties of the steps, and connect them.

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Been granted access to Deploy.
 - Created a component in Deploy. See [Creating a component in HCL DevOps Deploy on page 940](#).
 - Created a process for the component in Deploy. See [Creating a process in HCL DevOps Deploy on page 940](#).

About this task

When you open any process to configure, the process is displayed in the process editor. The process editor lists the plugins and steps. The required **Start** and **Finish** steps represent the beginning and the end of the process and steps are automatically placed on the design area.

You must provide the values for certain fields in the properties for the selected test step to run tests from Deploy.

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Components**.

Result

A list of components that are available in Deploy is displayed.

3. Select the component from the list in which you created the process.
4. Click the **Processes** tab.

Result

A list of processes that are available for the component is displayed.

5. Select the process from the list that you want to configure.

Result

The **Design** tab for the process is displayed.

6. Click **HCL DevOps Test Workbench**, and then **HCL DevOps Test Hub** from the left menu.
7. Drag the **Run HCL DevOps Test Hub test** step, and then drop it into the design area.



Note: The selected test must be placed between **Start** and **Finish** steps.

8. Specify the properties for the selected test by performing the following steps:


- a. Click the **Edit** icon.

Result

The **Edit Properties for Run HCL DevOps Test Hub test** dialog is displayed.



- b. Specify the properties for the selected test step by referring to the following table:


The following table lists the required fields that you must provide to run the test from Deploy:

Fields	Action
Name	Enter the name for the test step.
HCL DevOps Test Hub URL	Enter the URL of Test Hub.
Offline Token	Enter the offline token that you generated from Test Hub.
Team Space Name	Enter the name of the team space that contains the project.  Note: The license for the team space must be configured and you must be a member of that team space.
Project Name	Enter the name of the project that are available in the team space.
Repository Link	Enter the URL of the Git repository that you added to your project in Test Hub.
File Path	Enter the path of the test assets that you want to run. You can find the path of the test assets from the Execution page in Test Hub

The following table lists the optional fields that you can provide to run the test from Deploy:

Fields	Description
Branch Name	Use this field to enter the name of the branch in the Git repository where you stored test assets.
Custom Trust	Use this field to enter the trust store password if you have modified the password while creating the custom trust store.

Fields	Description
Store Password	 Note: If you have not modified the trust store password, you can retain the Custom Trust Store Password field blank.
Custom Trust Store Path	Use this field to enter the file path of your trust store followed by the file name if Test Hub uses an internal CA certificate and you have imported the certificate to a custom trust store.
Datasets	<p>Use this field to enter the path to the dataset if you want to replace the values of the dataset during a test run.</p> <p>You must ensure that both original and new datasets are in the same workspace and have the same column names. When you enter a value for the Datasets field, you must also include the path to the dataset. The path must be in the following format:</p> <pre>/project_name/ds_path/original_ds.csv:/project_name/ds_path/new_ds.csv</pre> <p> Note: You can override multiple datasets that are saved in a different project by adding multiple paths to the dataset separated by a semicolon.</p>
Environment	<p>Use this field to enter the environment details for the following types of test assets:</p> <ul style="list-style-type: none"> ▪ APISUITE ▪ APITEST ▪ APISTUB
Labels	<p>Use this field to add labels to test results when the test run is complete.</p> <p>You can add multiple labels to a test result separated by a comma.</p> <p>For example, <i>label1, label2</i></p> <p>After the test run is complete, then the values provided in the Labels field are displayed on the Results page of Test Hub.</p>
Classification Name	<p>Use this field to enter the name of the Classification if you created secrets for your project and to use Secrets at test run time.</p> <p>This field is mandatory only if you want to run the following types of test assets:</p>

Fields	Description
	<ul style="list-style-type: none"> ▪ APISUITE ▪ APITEST ▪ APISTUB
Show Hidden Properties	<p>Select this checkbox if you want to use an HTTP Proxy to connect to Test Hub.</p> <p>After you select the Show Hidden Properties checkbox, you can configure the HTTP Proxy by using the following fields:</p> <ul style="list-style-type: none"> ▪ HTTP Proxy Host: Use this field to enter the hostname of the HTTP proxy that you want to connect to Test Hub. ▪ HTTP Proxy Port: Use this field to enter the port number where the HTTP proxy is used to listen to both HTTP and HTTPS traffic. ▪ HTTP Proxy User name: Use this field to provide the username for the HTTP proxy. ▪ HTTP Proxy Password: Use this field to enter a valid password for the user that you specified in the HTTP Proxy User name field. <p> Note: You can use the Show Hidden Properties field only when you want to run the following types of test assets:</p> <ul style="list-style-type: none"> ▪ APISUITE ▪ APITEST ▪ APISTUB
Variables	<p>Use this field to enter the name of the variable and its value if your test requires variables during the test run time.</p> <p>You must enter the variables in the following format:</p> <pre>name_of_the_variable=value_of_the_varibale</pre> <p>You can add multiple variables to the test run separated by a semicolon.</p> <p>For example, <i>varname1=value1;varname2=value2</i></p>



Note: In addition to optional fields, you can also use the following fields from Deploy to configure your test run:

- **Working Directory**
- **Precondition**



- **Post Processing Script**
- **Use Impersonation**
- **Auth Token Restriction**

You can accept the default values or change the values based on your requirements. For more information about these fields, see the related links.

c. Click **OK** to save the properties for the test.

9. Click **Save** in the design area.

Results

You have configured the process for the component in Deploy.

What to do next

You must create a resource in Deploy. See [Creating a resource in HCL DevOps Deploy on page 946](#).

Creating a resource in HCL DevOps Deploy

You must create a resource to associate agents with components that you created in HCL DevOps Deploy (Deploy).

Before you begin

- You must be familiar with working with Deploy.
- You must have been granted access to Deploy.

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Resources**, and then click **Create Top-Level Group**.

Result

The **Create Resource** dialog is displayed.

3. Enter a name for the resource in the **Name** field.

4. Click **Save**.

Results

You have created the resource in Deploy.

What to do next

You must configure the resource. See [Configuring the resource in HCL DevOps Deploy on page 946](#).

Configuring the resource in HCL DevOps Deploy

You must configure the resource to add an agent and associate the agent with the component.

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Been granted access to Deploy.
 - Created a component in Deploy. See [Creating a component in HCL DevOps Deploy on page 940](#).
 - Created a resource in Deploy. See [Creating a resource in HCL DevOps Deploy on page 946](#).

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Resources**.

Result

A list of resources that are available in Deploy is displayed.

3. Perform the following steps to add an agent to the resource:

- a. Click the resource from the list for which you want to add an agent.
- b. Click the **Actions** icon from the last column, and then click **Add Agent**.
- c. Select the agent from the drop-down list.



Note: The **Name** field is auto populated with the name of the agent.

d. Click **Save**.

Result

The selected agent is added to the resource and you can view the status of the agent in the **Status** column.

4. Perform the following steps to add a component to the agent:

- a. Click the agent from the list for which you want to add a component.
- b. Click the **Actions** icon from the last column, and then click **Add Component**.
- c. Select the component from the drop-down list.



Note: The **Name** field is auto populated with the name of the component.

d. Click **Save**.

Result

The selected component is added to the agent.

Results

You have configured the resource in Deploy.

What to do next

You must create an application. See [Creating an application in HCL DevOps Deploy on page 948](#).

Creating an application in HCL DevOps Deploy

You must create an application to fetch all the components together that you want to deploy.

Before you begin

- You must be familiar with working with Deploy.
 - You must have been granted access to Deploy.
1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Applications**.
3. Click **Create Applications**, and then **New Applications**.
4. Enter a name for the application in the **Name** field.

Result

The **Environments** page for the application that you created is displayed.

Results

You have created the application in Deploy.

What to do next

You must configure the application. See [Configuring the application HCL DevOps Deploy on page 948](#).

Configuring the application HCL DevOps Deploy

You must configure the application to associate resources with environments and define processes to run test assets.

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Created a process for the component in Deploy. See [Creating a process in HCL DevOps Deploy on page 940](#).
 - Created a component in Deploy. See [Creating a component in HCL DevOps Deploy on page 940](#).
 - Created a process for the component in Deploy. See [Creating a process in HCL DevOps Deploy on page 940](#).

- Configure the process for the component in Deploy. See [Configuring the process in HCL DevOps Deploy on page 942](#).
- Created a resource in Deploy. See [Creating a resource in HCL DevOps Deploy on page 946](#).
- Configured the resource in Deploy. See [Configuring the resource in HCL DevOps Deploy on page 946](#).
- Created an application in Deploy. See [Creating an application in HCL DevOps Deploy on page 948](#).

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Applications**.

Result

A list of applications that are available in Deploy is displayed.

3. Click the application that you want to configure from the **Name** column.

Result

The **Environments** page for the selected application is displayed.

4. Perform the following steps to create an environment for the application that you selected:

- a. Click **Create Environment**.
- b. Enter a name for the environment in the **Name** field.
- c. Click **Save**.

5. Perform the following steps to configure resources to the environment:

- a. Click the environment that you created.
- b. Click **Add Base Resources**.

Result

A list of resources that are available in Deploy is displayed.

- c. Select the checkbox to add resources to the environment.
- d. Click **Save**.

Result

You can view the corresponding agent and the component that you added for the resource by using the **Expand** icon.

6. Perform the following steps to add the component to the application:

- a. Click **Applications**, and then select your application from the list.
- b. Click the **Components** tab, and then **Add Components**.
- c. Select the checkbox from the drop-down list to add components to the application.
- d. Click **Save**.

7. Perform the following steps to create a process for the application:

- a. Click the **Processes** tab, and then **Create Process**.
- b. Enter a name for the process in the **Name** field.
- c. Click **Save**.

Result

The **Design** tab for the process that you created is displayed.

8. Drag the component process listed under the **Component Process Steps** option from the left navigation pane and drop it into the design area.
9. Select the component process from the drop-down list in the **Operational (No Version Needed) Process** field.
10. Click **Save**.
11. Click the **Edit** icon, and then change the name of the properties.
12. Click **OK**, and then click **Save**.

Results

You have configured the application to run test assets from Deploy.

What to do next

You can run test assets from Deploy. See [Running tests from HCL DevOps Deploy on page 950](#).

Related information

[HCL DevOps Deploy Documentation](#)

Running tests from HCL DevOps Deploy

You can run test assets from HCL DevOps Deploy (Deploy) as a **Process** to test your application under test.

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Been granted access to Deploy.
 - Created a component in Deploy. See [Creating a component in HCL DevOps Deploy on page 940](#).
 - Created a process for the component in Deploy. See [Creating a process in HCL DevOps Deploy on page 940](#).
 - Configure the process for the component in Deploy. See [Configuring the process in HCL DevOps Deploy on page 942](#).
 - Created a resource in Deploy. See [Creating a resource in HCL DevOps Deploy on page 946](#).

- Configured the resource in Deploy. See [Configuring the resource in HCL DevOps Deploy on page 946](#).
- Created an application in Deploy. See [Creating an application in HCL DevOps Deploy on page 948](#).
- Configured the application in Deploy. See [Configuring the application HCL DevOps Deploy on page 948](#).
- Started the Deploy agent. For more information, refer to the [HCL Launch](#) documentation.

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Applications**.

Result

A list of applications that are available in Deploy is displayed.

3. Click the application that you configured for the test run.
4. Click the **Request Process** icon.

Result

The **Create Deployment** page is displayed.

5. Select the process that you created for the application which contains the test assets from the **Process** drop-down list.
6. Click **Next**, and then select the specific version of component, if any.



Note: If you have multiple versions for your component, then only you can choose the required version by clicking the **Add** option.

7. Click **Next**, and then verify the properties for the test run.
8. Click **Next**, and then verify that the status of the **Run Now** option is enabled.
You can disable the **Run Now** option, and then use the **Date**, **Time**, and **Recurrence Pattern** fields to run tests as **Process** at the scheduled time.
9. Click **Next**, and then verify the details about the **Process**.
10. Click **Submit Deployment** to initiate the test run.

Results

You have run the performance test assets as **Process** from Deploy.

What to do next

You can view the status of the process as follows:

- **Success:** When the test run is successful
- **Failed:** When the test run fails

You can view the details of the test run in the **Execution Log** section. The **Reports information** section on the `Log` file displays the names of the report along with its corresponding URLs.

Integration with IBM® DevOps Deploy

You can integrate HCL DevOps Test Hub (Test Hub) with IBM® DevOps Deploy (Deploy) to run test assets as a **Process** from Deploy.

When you use Deploy for automating the deployments of your applications to an environment during the development cycle, you can create test assets for your application in desktop clients. After you configure the project in Test Hub, you can then run those tests assets from Deploy.

You must use the DevOps Test Hub Deploy plugin to integrate Test Hub with Deploy. Integrating Test Hub with Deploy automates the process of test runs. If you have many tests to run at regular intervals, you can use Deploy to initiate test runs automatically.

The following table lists the tasks that you must perform to run performance test assets from Deploy:

Tasks	Go to...
Create any or all the following types of test assets in desktop clients to test your application: <ul style="list-style-type: none"> • API test assets • Functional test assets • Performance test assets 	<ul style="list-style-type: none"> • DevOps Test Performance documentation • DevOps Test UI documentation • DevOps Test Integrations and APIs documentation
Create a project, and then configure the repository in Test Hub.	Working with projects in a team space on page 1010
Install Deploy with at least one agent.	Installing the server System Requirements on page 83 for information about specific versions of Deploy requirements.
Install the DevOps Test Hub Deploy plugin on Deploy.	Installing the DevOps Test Hub Deploy plugin on page 953.
Create a Component in Deploy.	Creating a component in IBM DevOps Deploy on page 955.
Create a Process for the component in Deploy.	Creating a process in IBM DevOps Deploy on page 956.
Configure the Process that you created for the component in Deploy.	Configuring the process in IBM DevOps Deploy on page 957.

Tasks	Go to...
Create a Resource in Deploy.	Creating a resource in IBM DevOps Deploy on page 961.
Configure the Resource .	Configuring the resource in IBM DevOps Deploy on page 962.
Create an Application in Deploy.	Creating an application in IBM DevOps Deploy on page 963.
Configure the Application that you want to use to run test assets from Deploy.	Configuring the application in IBM DevOps Deploy on page 963.
Run the test assets from Deploy.	Running tests from IBM DevOps Deploy on page 965.

Installing the DevOps Test Hub Deploy plugin

You must install the DevOps Test Hub Deploy plugin to run test assets from IBM® DevOps Deploy (Deploy).

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Installed Deploy. For more information about the installation, refer to the [IBM Urban Code](#) documentation.
 - Been granted access to Deploy.
 - Downloaded the DevOps Test Hub Deploy plugin from the [HCL® License & Delivery portal](#).

For more information about specific versions of plug-in, see [Integration plugin compatibility matrix on page 924](#).

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Settings**.
3. Click **Automation Plugins** from the **Automation** pane.
4. Click **Load plugin**.
5. Click **Choose File** to locate and open the compressed DevOps Test Hub Deploy plugin file.



Remember: You must not extract the DevOps Test Hub Deploy plugin compressed file contents.

6. Click **Submit**.

Result

The installed plugin is displayed on the **Automation Plugins** tab.

Results

You have installed the DevOps Test Hub Deploy plugin on Deploy.

What to do next

You must create a **Component** in Deploy. See [Creating a component in IBM DevOps Deploy on page 955](#).

Using a custom trust store for IBM® DevOps Deploy

You can use a custom trust store in the DevOps Test Hub Deploy plugin file to establish a trusted and secure connection between the IBM® DevOps Deploy (Deploy) server and HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Saved the Certificate Authority (CA) assigned to Test Hub.



Note: You can contact the administrator of Test Hub if you do not have a copy of the CA.

- Installed Java on the computer from where you access the Test Hub UI.

About this task

If the SSL certificate assigned to Test Hub is signed by an internal Certified Authority (CA), then you must download and import the CA certificate to a custom trust store. You can then use the custom trust store in the **Run HCL DevOps Test Hub test** process step to establish a trusted and secure connection between the Deploy server and Test Hub.



Note: If the internal CA certificate is already imported to the default trust store that is used by the Deploy server, you need not use a custom trust store.

1. Locate the default trust store file (`cacerts` file) from the JRE directory.
For example: `C:\Program Files\Java\jdk-18.0.1.1\lib\security`
2. Copy the `cacerts` file to a location of your choice on your computer.
3. Place the CA assigned to Test Hub in the same location where you saved the `cacerts` file.
4. Run the following command from the command-line interface or terminal to import the CA of Test Hub to custom trust store:

```
keytool -import -trustcacerts -file {path to the downloaded CA with the file extension} -alias {label for the certificate} -keystore {path to the trust store cacerts file}
```

For example,

```
keytool -import -trustcacerts -file C:\Users\Desktop\cert\ca-master-hcl.crt -alias "otsca" -keystore
C:\Users\Desktop\cert\cacerts
```



Note: The default password of the trust store is *changeit*. It remains the same for the custom trust store. If you want to change the password, you can run the following command, and then enter the new password:

```
keytool -storepasswd -keystore {path to the trust store cacerts file}
```

For example, `keytool -storepasswd -keystore C:\Users\Desktop\cert\cacerts`

Results

You have successfully imported the downloaded CA to the custom trust store.

What to do next

You can then run the tests that are available in your Test Hub project from the Deploy server.

Creating a component in IBM® DevOps Deploy

You must create a component to include artifacts and processes in IBM® DevOps Deploy (Deploy). The artifacts include runnable files, images, databases, configuration instructions. Whereas the processes define the activities that components can perform.

Before you begin

- You must be familiar with working with Deploy.
- You must have been granted access to Deploy.

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Components**, and then click **Create Component**.
3. Enter a name for the component in the **Name** field.
4. Enter the details in the other optional fields based on your requirement, and then click **Save**.

Result

The component that you created is displayed.

Results

You have created the component in Deploy.

What to do next

You must create a process for the component in Deploy. See [Creating a process in IBM DevOps Deploy on page 956](#).

Creating a process in IBM® DevOps Deploy

You must create a process for the component to include step properties for the test that you want to run from IBM® DevOps Deploy (Deploy).

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Been granted access to Deploy.
 - Created a component in Deploy. See [Creating a component in IBM DevOps Deploy on page 955](#).

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Components**.

Result

A list of components that are available in Deploy is displayed.

3. Select the component from the list for which you want to create a process.
4. Click the **Processes** tab, and then click **Create Process**.

Result

The **Create Process** dialog is displayed.

5. Enter a name for the process in the **Name** field.
6. Select **Operational (No Version Needed)** from **Process Type** drop-down list.
7. Verify the **Default Working Directory** field.

The **Default Working Directory** field defines the location that the agent uses to run the process. The default value is `${p:resource/work.dir}/${p:component.name}`.

Where `${p:resource/work.dir}` is the default working directory for the agent and `${p:component.name}` is the name of the component.

8. Click **Save**.

Result

The process that you created is listed in the **Processes** tab and the **Design** tab for the process is displayed.

Results

You have created the process for the component in Deploy.

What to do next

You must configure the process in Deploy. See [Configuring the process in IBM DevOps Deploy on page 957](#).

Configuring the process in IBM® DevOps Deploy

You must configure the process that you created for the component to organize the steps in the process, specify the properties of the steps, and connect them in IBM® DevOps Deploy (Deploy).

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Been granted access to Deploy.
 - Created a component in Deploy. See [Creating a component in IBM DevOps Deploy on page 955](#).
 - Created a process for the component in Deploy. See [Creating a process in IBM DevOps Deploy on page 956](#).
 - Generated an offline user token from Test Hub. See [Generating an offline token on page 994](#).
 - Added the tests assets to the project on Test Hub.

About this task

When you open any process to configure, the process is displayed in the process editor. The process editor lists the plugins and steps. The required **Start** and **Finish** steps represent the beginning and the end of the process and steps are automatically placed on the design area.

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Components**.

Result

A list of components that are available in Deploy is displayed.

3. Select the component from the list in which you created the process.
4. Click the **Processes** tab.

Result

A list of processes that are available for the component is displayed.

5. Select the process from the list that you want to configure.

Result

The **Design** tab for the process is displayed.

6. Click **HCL DevOps Test Workbench**, and then **HCL DevOps Test Hub** from the left menu.
7. Drag the **Run HCL DevOps Test Hub test** step, and then drop it into the design area.



Note: The selected test must be placed between **Start** and **Finish** steps.

8. Specify the properties for the selected test by performing the following steps:


a. Click the **Edit** icon.

Result



The **Edit Properties for Run HCL DevOps Test Hub test** dialog is displayed.


b. Specify the properties for the selected test step by referring to the following tables:

The following table lists the required fields that you must provide to run the test from Deploy:

Fields	Action
Name	Enter the name for the test step.
HCL DevOps Test Hub URL	Enter the URL of Test Hub.
Offline Token	Enter the offline token that you generated from Test Hub.
Team Space Name	Enter the name of the team space that contains the project.  Note: The license for the team space must be configured and you must be a member of that team space.
Project Name	Enter the name of the project that are available in the team space.
Repository Link	Enter the URL of the Git repository that you added to your project in Test Hub.
File Path	Enter the path of the test assets that you want to run. You can find the path of the test assets from the Execution page in Test Hub

The following table lists the optional fields that you can provide to run the test from Deploy:

Fields	Description
Branch Name	Use this field to enter the name of the branch in the Git repository where you stored test assets.
Custom Trust Store Password	<p>Use this field to enter the trust store password if you have modified the password while creating the custom trust store.</p> <p> Note: If you have not modified the trust store password, you can retain the Custom Trust Store Password field blank.</p>
Custom Trust Store Path	Use this field to enter the file path of your trust store followed by the file name if Test Hub uses an internal CA certificate and you have imported the certificate to a custom trust store.
Datasets	<p>Use this field to enter the path to the dataset if you want to replace the values of the dataset during a test run.</p> <p>You must ensure that both original and new datasets are in the same workspace and have the same column names. When you enter a value for the Datasets field, you must also include the path to the dataset. The path must be in the following format:</p> <pre data-bbox="480 1146 1414 1209">/project_name/ds_path/original_ds.csv:/project_name/ds_path/new_ds.csv</pre> <p> Note: You can override multiple datasets that are saved in a different project by adding multiple paths to the dataset separated by a semicolon.</p>
Environment	<p>Use this field to enter the environment details for the following types of test assets:</p> <ul style="list-style-type: none"> ▪ APISUITE ▪ APITEST ▪ APISTUB
Labels	<p>Use this field to add labels to test results when the test run is complete.</p> <p>You can add multiple labels to a test result separated by a comma.</p> <p>For example, <i>label1, label2</i></p>

Fields	Description
	After the test run is complete, then the values provided in the Labels field are displayed on the Results page of Test Hub.
Classification Name	<p>Use this field to enter the name of the Classification if you created secrets for your project and to use Secrets at test run time.</p> <p>This field is mandatory only if you want to run the following types of test assets:</p> <ul style="list-style-type: none"> ▪ APISUITE ▪ APITEST ▪ APISTUB
Show Hidden Properties	<p>Select this checkbox if you want to use an HTTP Proxy to connect to Test Hub.</p> <p>After you select the Show Hidden Properties checkbox, you can configure the HTTP Proxy by using the following fields:</p> <ul style="list-style-type: none"> ▪ HTTP Proxy Host: Use this field to enter the hostname of the HTTP proxy that you want to connect to Test Hub. ▪ HTTP Proxy Port: Use this field to enter the port number where the HTTP proxy is used to listen to both HTTP and HTTPS traffic. ▪ HTTP Proxy User name: Use this field to provide the username for the HTTP proxy. ▪ HTTP Proxy Password: Use this field to enter a valid password for the user that you specified in the HTTP Proxy User name field. <p> Note: You can use the Show Hidden Properties field only when you want to run the following types of test assets:</p> <ul style="list-style-type: none"> ▪ APISUITE ▪ APITEST ▪ APISTUB
Variables	<p>Use this field to enter the name of the variable and its value if your test requires variables during the test run time.</p> <p>You must enter the variables in the following format:</p> <pre>name_of_the_variable=value_of_the_varibale</pre> <p>You can add multiple variables to the test run separated by a semicolon.</p>

Fields	Description
	For example, <i>varname1=value1;varname2=value2</i>



Note: In addition to optional fields, you can also use the following fields from Deploy to configure your test run:

- **Working Directory**
- **Precondition**
- **Post Processing Script**
- **Use Impersonation**
- **Auth Token Restriction**

You can accept the default values or change the values based on your requirements. For more information about these fields, see the related links.

c. Click **OK** to save the properties for the test.

9. Click **Save** in the design area.

Results

You have configured the process for the component in Deploy.

What to do next

You must create a resource in Deploy. See [Creating a resource in IBM DevOps Deploy on page 961](#).

Creating a resource in IBM® DevOps Deploy

You must create a resource to associate agents with components that you created in IBM® DevOps Deploy (Deploy).

Before you begin

- You must be familiar with working with Deploy.
- You must have been granted access to Deploy.

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Resources**, and then click **Create Top-Level Group**.

Result

The **Create Resource** dialog is displayed.

3. Enter a name for the resource in the **Name** field.

4. Click **Save**.

Results

You have created the resource in Deploy.

What to do next

You must configure the resource. See [Configuring the resource in IBM DevOps Deploy on page 962](#).

Configuring the resource in IBM® DevOps Deploy

You must configure the resource to add an agent and associate the agent with the component in IBM® DevOps Deploy (Deploy).

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Been granted access to Deploy.
 - Created a component in Deploy. See [Creating a component in IBM DevOps Deploy on page 955](#).
 - Created a resource in Deploy. See [Creating a resource in IBM DevOps Deploy on page 961](#).

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Resources**.

Result

A list of resources that are available in Deploy is displayed.

3. Perform the following steps to add an agent to the resource:

- a. Click the resource from the list for which you want to add an agent.
- b. Click the **Actions** icon from the last column, and then click **Add Agent**.
- c. Select the agent from the drop-down list.



Note: The **Name** field is auto populated with the name of the agent.

- d. Click **Save**.

Result

The selected agent is added to the resource and you can view the status of the agent in the **Status** column.

4. Perform the following steps to add a component to the agent:

- a. Click the agent from the list for which you want to add a component.
- b. Click the **Actions** icon from the last column, and then click **Add Component**.

- c. Select the component from the drop-down list.



Note: The **Name** field is auto populated with the name of the component.

- d. Click **Save**.

Result

The selected component is added to the agent.

Results

You have configured the resource in Deploy.

What to do next

You must create an application. See [Creating an application in IBM DevOps Deploy on page 963](#).

Creating an application in IBM® DevOps Deploy

You must create an application in IBM® DevOps Deploy (Deploy) to fetch all the components together that you want to deploy.

Before you begin

- You must be familiar with working with Deploy.
- You must have been granted access to Deploy.

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Applications**.
3. Click **Create Applications**, and then **New Applications**.
4. Enter a name for the application in the **Name** field.

Result

The **Environments** page for the application that you created is displayed.

Results

You have created the application in Deploy.

What to do next

You must configure the application. See [Configuring the application in IBM DevOps Deploy on page 963](#).

Configuring the application in IBM® DevOps Deploy

You must configure the application in IBM® DevOps Deploy (Deploy) to associate resources with environments and define processes to run test assets.

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Been granted access to Deploy.
 - Created a component in Deploy. See [Creating a component in IBM DevOps Deploy on page 955](#).
 - Created a process for the component in Deploy. See [Creating a process in IBM DevOps Deploy on page 956](#).
 - Configure the process for the component in Deploy. See [Configuring the process in IBM DevOps Deploy on page 957](#).
 - Created a resource in Deploy. See [Creating a resource in IBM DevOps Deploy on page 961](#).
 - Configured the resource in Deploy. See [Configuring the resource in IBM DevOps Deploy on page 962](#).
 - Created an application in Deploy. See [Creating an application in IBM DevOps Deploy on page 963](#).

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Applications**.

Result

A list of applications that are available in Deploy is displayed.

3. Click the application that you want to configure from the **Name** column.

Result

The **Environments** page for the selected application is displayed.

4. Perform the following steps to create an environment for the application that you selected:

- a. Click **Create Environment**.
- b. Enter a name for the environment in the **Name** field.
- c. Click **Save**.

5. Perform the following steps to configure resources to the environment:

- a. Click the environment that you created.

- b. Click **Add Base Resources**.

Result

A list of resources that are available in Deploy is displayed.

- c. Select the checkbox to add resources to the environment.

- d. Click **Save**.

Result

You can view the corresponding agent and the component that you added for the resource by using the **Expand** icon.

6. Perform the following steps to add the component to the application:
 - a. Click **Applications**, and then select your application from the list.
 - b. Click the **Components** tab, and then **Add Components**.
 - c. Select the checkbox from the drop-down list to add components to the application.
 - d. Click **Save**.
7. Perform the following steps to create a process for the application:
 - a. Click the **Processes** tab, and then **Create Process**.
 - b. Enter a name for the process in the **Name** field.
 - c. Click **Save**.

Result

The **Design** tab for the process that you created is displayed.

8. Drag the component process listed under the **Component Process Steps** option from the left navigation pane and drop it into the design area.
9. Select the component process from the drop-down list in the **Operational (No Version Needed) Process** field.
10. Click **Save**.
11. Click the **Edit** icon, and then change the name of the properties.
12. Click **OK**, and then click **Save**.

Results

You have configured the application to run test assets from Deploy.

What to do next

You can run test assets from Deploy. See [Running tests from IBM DevOps Deploy on page 965](#).

Running tests from IBM® DevOps Deploy

You can run test assets from IBM® DevOps Deploy (Deploy) as a **Process** to test your application under test.

Before you begin

- You must be familiar with working with Deploy.
- You must have performed the following tasks:
 - Been granted access to Deploy.
 - Created a component in Deploy. See [Creating a component in IBM DevOps Deploy on page 955](#).

- Created a process for the component in Deploy. See [Creating a process in IBM DevOps Deploy on page 956](#).
- Configure the process for the component in Deploy. See [Configuring the process in IBM DevOps Deploy on page 957](#).
- Created a resource in Deploy. See [Creating a resource in IBM DevOps Deploy on page 961](#).
- Configured the resource in Deploy. See [Configuring the resource in IBM DevOps Deploy on page 962](#).
- Created an application in Deploy. See [Creating an application in IBM DevOps Deploy on page 963](#).
- Configured the application in Deploy. See [Configuring the application in IBM DevOps Deploy on page 963](#).
- Started the Deploy agent. For more information, refer [Starting agents](#).

1. Log in to Deploy, if you are not already logged in.

Result

The Deploy dashboard is displayed.

2. Click **Applications**.

Result

A list of applications that are available in Deploy is displayed.

3. Click the application that you configured for the test run.

4. Click the **Request Process** icon.

Result

The **Run Application Process** page is displayed.

5. Select the process that you created for the application which contains the test assets from the **Process** drop-down list.

6. Click **Submit**.

Result

The Deploy dashboard shows the progress of the process request.

Results

You have run the performance test assets as **Process** from Deploy.

What to do next

After the process completes, you can view the status of the process displayed as follows:

- **Success:** When the test run is successful
- **Failed:** When the test run is failed

You can view the details of the test run in the **Execution Log** section. The **Reports information** section on the Log file displays the names of the report along with its corresponding URLs.

Integration with IBM® Engineering Test Management

As a *Project Owner*, you can integrate Engineering Test Management as a quality management solution with HCL DevOps Test Hub (Test Hub). You can run test assets that are available in a project from Engineering Test Management.

The following topics provides more information about the integration of Test Hub with Engineering Test Management.

Task flow for integration of IBM® Engineering Test Management

You must perform the following tasks in sequence to integrate HCL DevOps Test Hub (Test Hub) with Engineering Test Management. For more information about the tasks, refer to the links provided in the table.

Sl. No	Tasks	More information
1	Create a project in Test Hub.	Adding a project on page 1013
2	Install the required version of Engineering Test Management on your computer.	Engineering Lifecycle Management Documentation
3	Download the Engineering Test Management adapter, and then extract the contents of a compressed file. For more information about specific versions of plugin, see Integration plugin compatibility matrix on page 924 .	HCL® License & Delivery portal
4	Add the adapter files that you downloaded to the installation directory of Engineering Test Management.	Adding the adapter files to IBM Engineering Test Management on page 967
5	Configure the details of Engineering Test Management on Test Hub.	Configuring IBM Engineering Test Management on DevOps Test Hub on page 969
6	Create a test script and a test case on Engineering Test Management to run the test assets.	Preparing test cases in IBM Engineering Test Management on page 970
7	Run the test assets from Engineering Test Management.	Running tests from IBM Engineering Test Management on page 972

Adding the adapter files to IBM® Engineering Test Management

You must add the adapter files to the installation directory to run test assets of HCL DevOps Test Hub (Test Hub) from Engineering Test Management.

Before you begin

You must have completed the following tasks:

- Installed the latest version of Engineering Test Management from the [Jazz](#) portal on your computer. See [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).
- Downloaded the adapter (a compressed file) for Engineering Test Management from the [HCL® License & Delivery](#) portal.

For more information about specific versions of plugin, see [Integration plugin compatibility matrix on page 924](#).

About this task

You must add the adapter files to the location where you installed Engineering Test Management to run test assets of Test Hub.

1. Extract the compressed file that you downloaded.
2. Copy the `etm-adapter-updatesite` folder.
3. Go to the location where you installed Engineering Test Management on your computer.
4. Place the `etm-adapter-updatesite` folder in the `sites` folder of the installation directory of Engineering Test Management.

For example, `D:\ELM\JazzTeamServer\server\conf\qm\sites`.

5. Copy the `ots-profile.ini` file from the `provision_profiles` folder that is available in the extracted file.
6. Place the `ots-profile.ini` file in the `provision_profiles` folder of the installation directory of Engineering Test Management.

For example, `D:\ELM\JazzTeamServer\server\conf\qm\provision_profiles`

Results

You have added the adapter files to Engineering Test Management.

What to do next

You must perform the following tasks:

- You must reset the Engineering Test Management server from the web console by using the **Request Server Reset** option.

If you have administrator privileges, then you can access the **Request Server Reset** option from the following URL:

```
https://<hostname>:<portnumber>/qm/admin?  
internal#action=com.ibm.team.repository.admin.serverReset
```



Note: You must replace the following values with the actual values in the URL:

- `<hostname>` with the hostname that you used when you installed Engineering Test Management.
 - `<portnumber>` with the port number that is associated with an IP address of the hostname.
- Restart the Engineering Test Management server.
 - Configure the details of Engineering Test Management on Test Hub to establish a connection between the two products. See [Configuring IBM Engineering Test Management on DevOps Test Hub on page 969](#).

Configuring IBM® Engineering Test Management on DevOps Test Hub

You must configure Engineering Test Management on HCL DevOps Test Hub (Test Hub) to establish a connection between the two products.

Before you begin

You must have completed the following tasks:

- Granted sufficient privileges to access **Project area** on Engineering Test Management.
- Created a project area in Engineering Test Management.
- Created a project in Test Hub.

About this task

If you have multiple instances of Engineering Test Management, then you can configure all those instances on Test Hub. If you have only one instance of Engineering Test Management, you can configure the same instance multiple times. However, you must choose a different **Project area** during the configuration of Engineering Test Management.

1. Log in to Test Hub, if you are not already logged in.
2. Open a project from the team space.
3. Click **Manage > Integration**.
4. Click the **TEST MANAGEMENT** tab, and then click **Add IBM ETM adapter**.
5. Enter the following information in the **Add IBM ETM adapter** dialog:

Fields	Actions
Server URL	Enter the URL of Engineering Test Management. For example, <code>https://<hostname>:<portnumber>/qm</code>
Username	Enter the username associated with Engineering Test Management.

Fields	Actions
Password	Enter the password associated with the username of Engineering Test Management.

6. Click **Validate**.
7. Select the project name that you created in Engineering Test Management from the **Project area** list.
8. Enter a name for the Engineering Test Management adapter in the **Adapter name** field.



Note: You cannot provide an adapter name that already exists for the **Project area**.

9. Click **Save**.

Results

You have configured the details of Engineering Test Management on Test Hub.

What to do next

You must create a test script and test case on Engineering Test Management to run the test assets of Test Hub. See [Preparing test cases in IBM Engineering Test Management on page 970](#).

Preparing test cases in IBM® Engineering Test Management

You must create a test script and a test case on Engineering Test Management to run the test assets.



Before you begin

You must have completed the following tasks:

- Added the adapter files to the installation directory of Engineering Test Management. See [Adding the adapter files to IBM Engineering Test Management on page 967](#).
- Configured the details of Engineering Test Management on HCL DevOps Test Hub (Test Hub). See [Configuring IBM Engineering Test Management on DevOps Test Hub on page 969](#).
- Generated an offline user token from Test Hub. See [Generating an offline token on page 994](#).
- Copied the following values to be used during the preparation of a test script and a test case:
 - Branch name of the Git repository
 - Asset ID of the test that you want to run
 - **Optional:** Name of the environment if you want to run the API test suite.

1. Log in to Engineering Test Management, if you are not already logged in.
2. Open your project area from where you want to run the test assets.
3. Click **Construction**, and then click **Test Script** from the drop-down list.

4. Enter a name for the test script in the **Test Script** field.
5. Select **HCL DevOps Test Hub** from the **Type** drop-down list.
6. Enter the following test information:

Fields	Actions
Offline Token	Enter the offline token that you generated from Test Hub.
Branch	Enter the name of the branch where your test assets are stored.
Asset ID	Enter an asset ID of the test asset that is available on the Execution page of Test Hub.  Tip: You can hover over the name of the test asset to view and copy the asset ID.
Environment	Enter a name of the API test environment.  Remember: You must use this field only if you want to run the API test suites.

7. Click **Save**.
8. Perform the following steps to add a test script to a test case:

- a. Click **Construction > Test Case**.

Result

The **Test Cases** page is displayed.

- b. Enter a name for the test case in the **Test Case** field.
- c. Click **Test Scripts** from the left navigation pane.
- d. Click the **Plus** icon from the **Test Scripts** panel.

Result

The **Add Test Scripts** dialog is displayed.

- e. Select the checkbox to add the test script that you created to the test case, and then click **Add and Close**.
- f. Click **Save**.

Results

You have prepared Engineering Test Management.

What to do next

You can run the test assets of Test Hub from Engineering Test Management. See [Running tests from IBM Engineering Test Management on page 972](#).

Running tests from IBM® Engineering Test Management

After you create a test case in Engineering Test Management, you can run the test. The test case includes the test scripts that contains test assets of HCL DevOps Test Hub (Test Hub).

Before you begin

You must have performed the following tasks:


- Prepared Engineering Test Management to run test assets of Test Hub. See [Preparing test cases in IBM Engineering Test Management on page 970](#).
- Configured the details of Engineering Test Management on Test Hub. See [Configuring IBM Engineering Test Management on DevOps Test Hub on page 969](#).

1. Log in to Engineering Test Management.
2. Open the test case that you created in Engineering Test Management.

3. Click the **Play** icon  , and then click **Run**.

Result

The **Run Test Case** dialog is displayed.

4. Click the **ChangeMachine** icon  next to the **Machine** field to select the adapter that you configured on Test Hub.
5. Select the required adapter name from the list, and then click **OK**.
6. Click **Finish** to start the test run.

The Engineering Test Management dashboard displays the progress of the test case run.

Results

You have run the test assets of Test Hub from Engineering Test Management.

What to do next

You can perform the following tasks:

- Click **Show Result** to view information about the test results from Engineering Test Management when the test run is complete.
- Click **Link to Result** from **Results details** to view the test results from the **Results** page of Test Hub. See [Test results and reports on page 789](#).

Deleting an IBM® Engineering Test Management adapter

You might want to delete an IBM® Engineering Test Management adapter when it is no longer needed.


Before you begin

You must have completed the following tasks:

- Been assigned the role of *Owner* of a project or *Owner* of the project in HCL DevOps Test Hub (Test Hub).
- Configured the details of Engineering Test Management on Test Hub. See [Configuring IBM Engineering Test Management on DevOps Test Hub on page 969](#).

1. Log in to Test Hub and open the team space that contains your project.
2. Click **Manage > Integration**.
3. Click the **TEST MANAGEMENT** tab.

You can view the Engineering Test Management adapters that are configured with the project on Test Hub.

4. Identify the Engineering Test Management adapter that you want to delete, and then click the **Menu** icon .
5. Click **Delete**.



Note: If any of the test runs are in progress on Engineering Test Management, then those runs are aborted.

6. Click **Confirm** to delete the Engineering Test Management adapter from the project.

Results

You have deleted the Engineering Test Management adapter from the project on Test Hub.

Integration with Jenkins

You can use the HCL DevOps Test Hub (Test Hub) plugin to integrate Test Hub with Jenkins. After you complete the integration, you can run tests assets that are available in a project of Test Hub from a Jenkins server.

You must have installed Jenkins on your computer to install the Test Hub Jenkins plugin. You can run the following command to start the Jenkins server to support UTF-8 character sets:

```
java -Dfile.encoding=UTF8 -jar jenkins.war
```

You can use either the Jenkins **Freestyle** project or the **Pipeline** project to run test assets from Jenkins. With **Freestyle** project, you can create build step from the Jenkins UI to run the test assets. Whereas the **Pipeline** project uses a simple text scripts based on the Groovy programming language. You can define pipeline scripts during the configuration of the **Pipeline** project by using any of the following methods:

- **Pipeline script:** In this method, you can directly enter a script in the field provided within the Jenkins UI or you can select a sample **Pipeline script** from the drop-down list.
- **Pipeline script from SCM:** In this method, you can create a script with a Groovy editor and then commit the script file into the Git repository.

For more information about the Jenkins Pipeline, refer to the [Jenkins](#) documentation.

Refer to the following topics to learn more about integrating Jenkins with Test Hub.

Task flows for running test assets from Jenkins

You can perform certain tasks to run test assets from the Jenkins **Freestyle** project or the **Pipeline** project.

The following table lists the task flows for running test assets from the Jenkins **Freestyle** project:

Tasks	Go to...
Install the Test Hub Jenkins plugin.	Installing the Test Hub Jenkins plugin on page 975
Import the CA certificate to your custom trust store.	Using a custom trust store on page 975
Configure the Freestyle project.	Configuring the Freestyle project on page 977
Run Test Hub test assets on Jenkins.	Running tests from Jenkins on page 987

You can perform the following tasks to run test assets from the Jenkins **Pipeline** project by using the sample script:

Tasks	Go to...
Install the Test Hub Jenkins plugin.	Installing the Test Hub Jenkins plugin on page 975
Import the CA certificate to your custom trust store.	Using a custom trust store on page 975
Create a pipeline script from the Jenkins UI.	Creating a pipeline script from Jenkins on page 981
Configure the Pipeline project.	Configuring the Pipeline project by using the sample script on page 985
Run Test Hub test assets on Jenkins.	Running tests from Jenkins on page 987

You can perform the following tasks to run test assets from the Jenkins **Pipeline** project by using the script from Source Code Management (SCM):

Tasks	Go to...
Install the Test Hub Jenkins plugin.	Installing the Test Hub Jenkins plugin on page 975
Import the CA certificate to your custom trust store.	Using a custom trust store on page 975
Create a pipeline script and commit the script to the Git repository.	For more information, refer to the Jenkins documentation.

Tasks	Go to...
Configure the Pipeline project.	Configuring the Pipeline project by using the script from SCM on page 986
Run Test Hub test assets on Jenkins.	Running tests from Jenkins on page 987

Installing the Test Hub Jenkins plugin

Before you can run the tests that are available in your HCL DevOps Test Hub (Test Hub) project on the Jenkins server, you must install the Test Hub Jenkins plugin on the Jenkins server.

Before you begin

You must have completed the following tasks:

- Installed Jenkins on your computer. For more information about installing Jenkins, refer to the [Jenkins](#) documentation.
- Downloaded the Test Hub Jenkins plugin from the [HCL® License & Delivery portal](#).

For more information about specific versions of plugin, see [Integration plugin compatibility matrix on page 924](#).

1. Log in to the Jenkins server.

Result

The Jenkins dashboard is displayed.

2. Click **Manage Jenkins > Manage Plugins**.
3. Click the **Advanced** tab.
4. Click **Choose File** from the **Upload plugin** section to locate and **Open** the Test Hub Jenkins plugin.
5. Click **Upload**.

Result

The Test Hub Jenkins plugin is displayed in the **Installed** tab.

Results

You have installed the Test Hub Jenkins plugin on the Jenkins server.

What to do next

You must import the CA certificate to your custom trust store to establish a trusted and secure connection between the Jenkins server and Test Hub. See [Using a custom trust store on page 975](#).

Using a custom trust store

You must use a custom trust store in the Jenkins build to establish a trusted and secure connection between the Jenkins server and HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Saved the Certificate Authority (CA) assigned to Test Hub.



Note: You can contact the administrator of Test Hub if you do not have a copy of the CA.

- Installed Java on the computer from where you access the Test Hub UI.

About this task

If the SSL certificate assigned to Test Hub is signed by an internal CA, then you must download and import the CA to a custom trust store. You can then use the custom trust store in the Jenkins build step to establish a trusted and secure connection between the Jenkins server and Test Hub.



Note: If the internal CA is already imported to the default trust store that is used by the Jenkins server, you need not use a custom trust store.



Restriction:

When you use Red Hat Enterprise Linux (RHEL) operating systems for Jenkins, you must run the Jenkins service with a user who has access to the custom trust store path to utilize the custom trust store feature. To change the Jenkins user, you must open the `/etc/sysconfig/jenkins` file and set the `JENKINS_USER` to the user who has access to the custom trust store path.

You can then run the following commands to set `JENKINS_USER` to another user who has access to the path of the custom trust store:

```
$JENKINS_USER= <username>
```

For example, `$JENKINS_USER= <user1>`



Note: You must ensure that the user account is available in the `/etc/passwd` file.

You can then run the following commands to change the ownership of the Jenkins folder:


```
chown -R username:username /var/lib/jenkins
```

```
chown -R username:username /var/cache/jenkins
```

```
chown -R username:username /var/log/jenkins
```

For example,

```
chown -R user1:user1 /var/lib/jenkins
```

 `chown -R user1:user1 /var/cache/jenkins`

`chown -R user1:user1 /var/log/jenkins`

After the change of ownership is complete, you can run the following command to restart the Jenkins server:

```
/etc/init.d/jenkins restart
```

1. Locate the default trust store file (`cacerts` file) from the JRE directory.
For example: `C:\Program Files\Java\jdk-18.0.1.1\lib\security`
2. Copy the `cacerts` file to a location of your choice on your computer.
3. Place the CA assigned to Test Hub in the same location where you saved the `cacerts` file.
4. Run the following command from the command-line interface or terminal to import the CA of Test Hub to custom trust store:

```
keytool -import -trustcacerts -file {path to the downloaded CA with the file extension} -alias {label for the certificate} -keystore {path to the trust store cacerts file}
```

For example,

```
keytool -import -trustcacerts -file C:\Users\Desktop\cert\ca-master-hcl.crt -alias "otsca" -keystore C:\Users\Desktop\cert\cacerts
```



Note: The default password of the trust store is *changeit*. It remains the same for the custom trust store. If you want to change the password, you can run the following command, and then enter the new password:

```
keytool -storepasswd -keystore {path to the trust store cacerts file}
```

For example, `keytool -storepasswd -keystore C:\Users\Desktop\cert\cacerts`

Results

You have successfully imported the downloaded CA to the custom trust store.

What to do next

You can run test assets either from **Freestyle** or **Pipeline** project from the Jenkins server.

Configuring the Freestyle project

You must configure a **Freestyle** project to add a build step, and then run tests that are available in your HCL DevOps Test Hub (Test Hub) project from Jenkins.

Before you begin


You must have completed the following tasks:



- Installed the Test Hub Jenkins plugin on the Jenkins server. See [Installing the Test Hub Jenkins plugin on page 975](#).
 - Created a Jenkins **Freestyle** project.
 - Generated an offline user token from Test Hub. See [Managing access to DevOps Test Hub on page 994](#).
1. Open your Jenkins **Freestyle** project from the list, and then click **Configure**.
 2. Click the **Build** tab, and then click **Add build step**.
 3. Select the **HCL DevOps Test Hub test** option from the drop-down list.
 4. Provide the details about the test run for the fields in the following table:


The following table lists the required fields that you must provide to run the test from Jenkins:


Fields	Action
Name	Enter a name for the Jenkins build step.
Server URL	Enter the URL of Test Hub. The format of the URL is as follows: <i>https://hostname</i> .
Offline Token	Enter the offline user token that you generated from Test Hub.


The following table lists the optional fields that you can provide to run the test from Jenkins:

Fields	Descriptions
Asset Type	Use this field to select the test asset type that you want to run from the Asset Type drop-down list.
Branch	Use this field to select the branch of the Git repository from the drop-down list. The Branch drop-down list displays the branches that are available in the corresponding project of Test Hub.  Note: After you select the branch from the Branch drop-down list, if you want to change the URL of Test Hub, offline user token, or the project in the Build step, then you must close the build step. Then you must click Add build step , select Run HCL DevOps Test Hub test , and then enter the details.
Datasets	Use this field to enter the path to the dataset if you want to replace the values of the dataset during a test run.

Fields	Descriptions
	<p>You must ensure that both original and new datasets are in the same workspace and have the same column names. When you enter a value for the Datasets field, you must also include the path to the dataset. The path must be in the following format:</p> <pre data-bbox="410 415 1419 443">/project_name/ds_path/original_ds.csv:/project_name/ds_path/new_ds.csv</pre> <p> Note: You can override multiple datasets that are saved in a different project by adding multiple paths to the dataset separated by a semicolon.</p>
Labels	<p>Use this field to add labels to test results when the test run is complete.</p> <p>You can add multiple labels to a test result separated by a comma.</p> <p>For example, <i>label1, label2</i></p> <p>After the test run is complete, then the values provided in the Labels field are displayed on the Results page of Test Hub.</p>
Project	<p>Use this field to select a project from the drop-down list.</p> <p>The Project drop-down list displays the projects that are available in the corresponding team space of Test Hub.</p> <p> Note: The Project drop-down list displays the projects where you are an <i>Owner</i> or a <i>member</i> of Test Hub. You must be an <i>Owner</i> or a <i>Tester</i> of the project that is available in the team spaces to run the tests from the Jenkins server.</p>
Secrets	<p>Use this field to select the Secret from the Classification drop-down list if you created secrets for your project to use Secrets at test run time.</p> <p>The Secrets field is mandatory only if you select one of the following options in the Asset Type field:</p> <ul style="list-style-type: none"> ◦ APISUITE ◦ APITEST ◦ APISTUB ◦ POSTMAN <p>The following message is displayed when you select any other asset type than preceding list:</p> <pre data-bbox="410 1791 1419 1854">You can select Secrets only if you are running tests of type APISUITE, APITEST, APISTUB, or POSTMAN.</pre>

Fields	Descriptions
Team Space	<p>Use this field to select the name of the team space from the drop-down list.</p> <p>The Team Space drop-down list displays the names of the team spaces if you are an <i>Owner</i> or a <i>member</i> of the team space.</p>
Test	<p>Use this field to select the required test that you want to run from the drop-down list.</p> <p>The Tests drop-down list displays the available test assets from the corresponding branch in the following format:</p> <pre>Project_name [Path: the path of the test assets] [Repo: URL of the Git repository that the test belongs to]</pre> <p> Note: After you select the test details from the Tests drop-down list if you want to change the URL of Test Hub, offline user token, or the project in the Jenkins Build step, then you must delete the build step and add a new build step.</p>
Test Environment	<p>Use this field to select an environment from the drop-down list.</p> <p>The Test Environment field displays the available test environments from Test Hub.</p> <p>The Test Environment field is mandatory only if you select one of the following options in the Asset Type field:</p> <ul style="list-style-type: none"> ◦ APISUITE ◦ APITEST ◦ APISTUB ◦ POSTMAN <p>The following message is displayed when you select any other asset type than preceding list:</p> <pre>You can select Test Environment only if you are running tests of type APISUITE, APITEST, APISTUB, or POSTMAN.</pre>
Use Custom Trust Store	<p>Select this checkbox if you used an internal CA certificate and imported the certificate to the custom trust store. You can then enter the file path of your trust store followed by the file name in the Custom Trust Store Path field.</p> <p>Select the Use Custom Password for the Trust Store checkbox if you modified the trust store password, and then enter the new password in the Custom Trust Store Password field.</p>

Fields	Descriptions
	 Note: If you did not modify the trust store password, you must clear the Use Custom Password for the Trust Store checkbox.
Variables	<p>Use this field to enter the name of the variable and its value if your test requires variables during the test run time.</p> <p>You must enter the variables in the following format:</p> <pre>name_of_the_variable=value_of_the_varibale</pre> <p>You can add multiple variables to the test run separated by a semicolon.</p> <p>For example, <i>varname1=value1;varname2=value2</i></p>

 **Note:** You can run multiple tests sequentially in the same job by adding multiple build steps and providing details for the tests that you want to run.

5. Click **Save** to save the build step.

Results

You have configured the **Freestyle** project by adding the build step.

What to do next

You can run test assets from the Jenkins server. See [Running tests from Jenkins on page 987](#).

Creating a pipeline script from Jenkins

You must create a pipeline script to run test assets that are available in the HCL DevOps Test Hub (Test Hub) project from the Jenkins **Pipeline** project.

Before you begin

You must have completed the following tasks:


- Installed the Test Hub Jenkins plugin on the Jenkins server. See [Installing the Test Hub Jenkins plugin on page 975](#).
- Generated an offline user token from Test Hub. See [Managing access to DevOps Test Hub on page 994](#).
- Created a Jenkins **Pipeline** project.


1. Open your Jenkins **Pipeline** project from the list.
2. Click **Configure**, and then select the **Pipeline** tab.



3. Click **Pipeline Syntax** to generate a pipeline script.
4. Select the **step: General Build Step** option from the drop-down list in the **Sample Step** field.
5. Select the **Run HCL DevOps Test Hub test** option from the drop-down list in the **Build Step** field.
6. Provide the details about the test run for the fields in the following table:

The following table lists the required fields that you must provide to run the test from Jenkins:

Fields	Action
Name	Enter a name for the Jenkins build step.
Server URL	Enter the URL of Test Hub. The format of the URL is as follows: <i>https://hostname</i> .
Offline Token	Enter the offline user token that you generated from Test Hub.

Field	Description
Asset Type	Use this field to select the test asset type that you want to run from the Asset Type drop-down list.
Branch	Select the branch from the drop-down list. The Branch drop-down list displays the branches that are available in the corresponding project of Test Hub.
Datasets	Use this field to enter the path to the dataset if you want to replace the values of the dataset during a test run. You must ensure that both original and new datasets are in the same workspace and have the same column names. When you enter a value for the Datasets field, you must also include the path to the dataset. The path must be in the following format: <code>/project_name/ds_path/original_ds.csv:/project_name/ds_path/new_ds.csv</code>  Note: You can override multiple datasets that are saved in a different project by adding multiple paths to the dataset separated by a semicolon.
Labels	Use this field to add labels to test results when the test run is complete. You can add multiple labels to a test result separated by a comma.

Field	Description
	<p>For example, <i>label1</i>, <i>label2</i></p> <p>After the test run is complete, then the values provided in the Labels field are displayed on the Results page of Test Hub.</p>
Project	<p>Use this field to select a project from the drop-down list.</p> <p>The Project drop-down list displays the projects that are available in the corresponding team space of Test Hub.</p> <p> Note: The Project drop-down list displays the projects where you are an <i>Owner</i> or a <i>member</i> of Test Hub. You must be an <i>Owner</i> or a <i>Tester</i> of the project that is available in the team spaces to run the tests from the Jenkins server.</p>
Secrets	<p>Use this field to select the Secret from the Classification drop-down list if you created secrets for your project to use Secrets at test run time.</p> <p>The Secrets field is mandatory only if you select one of the following options in the Asset Type field:</p> <ul style="list-style-type: none"> ◦ APISUITE ◦ APITEST ◦ APISTUB ◦ POSTMAN <p>The following message is displayed when you select any other asset type than preceding list:</p> <p>You can select Secrets only if you are running tests of type APISUITE, APITEST, APISTUB, or POSTMAN.</p>
Team Space	<p>Use this field to select the name of the team space from the drop-down list.</p> <p>The Team Space drop-down list displays the names of the team spaces if you are an <i>Owner</i> or a <i>member</i> of the team space.</p>
Test	<p>Use this field to select the required test that you want to run from the drop-down list.</p> <p>The Tests drop-down list displays the available test assets from the corresponding branch in the following format:</p> <p>Project_name [Path: the path of the test assets] [Repo: URL of the Git repository that the test belongs to]</p>

Field	Description
	 Note: After you select the test details from the Tests drop-down list if you want to change the URL of Test Hub, offline user token, or the project in the Jenkins Build step , then you must delete the build step and add a new build step.
Test Environment	<p>Use this field to select an environment from the drop-down list.</p> <p>The Test Environment field displays the available test environments from Test Hub.</p> <p>The Test Environment field is mandatory only if you select one of the following options in the Asset Type field:</p> <ul style="list-style-type: none"> ◦ APISUITE ◦ APITEST ◦ APISTUB ◦ POSTMAN <p>The following message is displayed when you select any other asset type than preceding list:</p> <pre>You can select Test Environment only if you are running tests of type APISUITE, APITEST, APISTUB, or POSTMAN.</pre>
Use Custom Trust Store	<p>Select this checkbox if you used an internal CA certificate and imported the certificate to the custom trust store. You can then enter the file path of your trust store followed by the file name in the Custom Trust Store Path field.</p> <p>Select the Use Custom Password for the Trust Store checkbox if you modified the trust store password, and then enter the new password in the Custom Trust Store Password field.</p>  Note: If you did not modify the trust store password, you must clear the Use Custom Password for the Trust Store checkbox.
Variables	<p>Use this field to enter the name of the variable and its value if your test requires variables during the test run time.</p> <p>You must enter the variables in the following format:</p> <pre>name_of_the_variable=value_of_the_varibale</pre> <p>You can add multiple variables to the test run separated by a semicolon.</p>

Field	Description
	For example, <code>varname1=value1;varname2=value2</code>

- Click **Generate Pipeline Script** and copy the script that you created for the build step to use it in the sample script.

Results

You have created the pipeline script.

What to do next

You can configure the **Pipeline** project by using the **Pipeline script** option. See [Configuring the Pipeline project by using the sample script on page 985](#).

Configuring the Pipeline project by using the sample script

You can configure a **Pipeline** project directly from the Jenkins UI by using the sample pipeline scripts to run test assets from Jenkins.

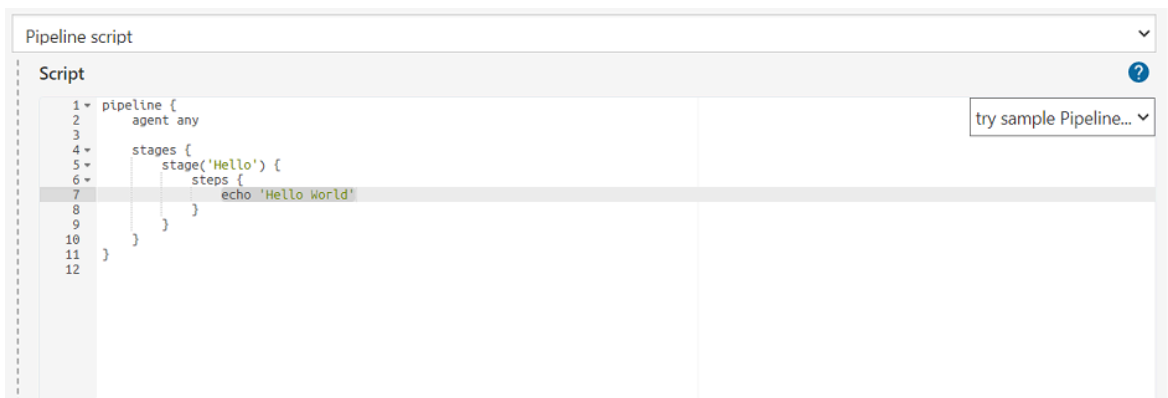
Before you begin

You must have completed the following tasks:

- Installed the Test Hub Jenkins plugin on the Jenkins server. See [Installing the Test Hub Jenkins plugin on page 975](#).
- Created a pipeline script. See [Creating a pipeline script from Jenkins on page 981](#).

- Open your Jenkins **Pipeline** project from the list.
- Click **Configure**, and then select the **Pipeline** tab.
- Select the **Pipeline script** option in the **Definition** field.
- Select the sample script from the drop-down list.

For example, if you select `Hello World` from the drop-down list, then the sample script is displayed as follows:



```

1 pipeline {
2   agent any
3
4   stages {
5     stage('Hello') {
6       steps {
7         echo 'Hello World'
8       }
9     }
10  }
11 }
12

```

5. Specify the script that you created in place of `echo 'Hello Word'`.



The screenshot shows the Jenkins Pipeline script editor. The script is as follows:

```

1 pipeline {
2   agent any
3
4   stages {
5     stage('Hello') {
6       steps {
7         step([$class: 'ServerTestRunner', assetType: 'COMPOUND', branches: '', customPasswordUsage: false, customStoreUsage: false, er
8       ]
9     }
10  }
11 }
12

```

There is a dropdown menu on the right side of the editor with the text "try sample Pipeline..." and a question mark icon.



Note: You can provide a different name for the `stage` in the script.

Results

You have configured the **Pipeline** project by using the **Pipeline script** option.

What to do next

You can run test assets from the Jenkins server. See [Running tests from Jenkins on page 987](#).

Configuring the Pipeline project by using the script from SCM

You can configure the **Pipeline** project to use the pipeline script from the Source Code Management (SCM), and then run test assets that are available in the HCL DevOps Test Hub (Test Hub) project from the Jenkins **Pipeline** project.

Before you begin

You must have completed the following tasks:

- Installed the Test Hub Jenkins plugin on the Jenkins server. See [Installing the Test Hub Jenkins plugin on page 975](#).
- Created a pipeline script and committed the script to the Git repository. For more information, refer to the [Jenkins](#) documentation.
- Been granted administrator permission of the Jenkins server to add credentials into Jenkins.
- Added your GitHub credentials into Jenkins. For more information about adding global credentials, refer to the [Jenkins](#) documentation.
- Created a Jenkins **Pipeline** project.

About this task

If your pipeline scripts are complex, you can then write the script outside of Jenkins UI. You must commit that script file into your Git repository. During the **Pipeline** build process, Jenkins checks out the script file from the Git repository and runs your **Pipeline** project.

1. Open your Jenkins **Pipeline** project from the list.
2. Click **Configure**, and then select the **Pipeline** tab.
3. Select the **Pipeline script from SCM** option in the **Definition** field.
4. Select the **Git** option from the drop-down list in the **SCM** field.
5. Enter the URL of the Git repository in the **Repository URL** field where you stored your pipeline script.
6. Select the credentials of the Git repository from the drop-down list.
7. **Optional:** Enter the branch name of the Git repository in the **Branch Specifier** field.
8. Enter the path of the script that you stored in the Git repository in the **Script path** field.
9. Click **Save**.

Results

You have configured the **Pipeline** project by using the **Pipeline script from the SCM** option.

What to do next

You can run test assets from the Jenkins server. See [Running tests from Jenkins on page 987](#).

Running tests from Jenkins

You can run test assets either from the Jenkins **Freestyle** or the **Pipeline** project on the Jenkins server to test an application under test.

Before you begin

You must have completed the following tasks:

- Configured the **Freestyle** project, if you want to run test assets from the **Freestyle** project. See [Configuring the Freestyle project on page 977](#).
- Configured the **Pipeline** project either by using the **Pipeline script** or **Pipeline script from SCM**, if you want to run test assets from the **Pipeline script** project. See [Configuring the Pipeline project by using the sample script on page 985](#) or [Configuring the Pipeline project by using the script from SCM on page 986](#).
- Downloaded the latest **Test Results Analyzer** plugin from the [Jenkinsci](#) repository. For more information about the **Test Results Analyzer** plugin, refer to the [Jenkins](#) documentation.
- Installed the **Test Results Analyzer** plugin on the Jenkins server. For more information about installing the plugin, refer to the [Jenkins](#) documentation.

About this task

After the test run is complete, you can view the test results in a tabular format when you use the Jenkins **Test Results Analyzer** plugin. With a TRA-based report, you can filter the results based on the status of the build. The status of the

builds can be `PASSED`, `FAILED`, and `SKIPPED`. You can also identify the failed build from the TRA-based report and take the required action to resolve the build errors. The TRA report also represents the build information in **Line**, **Bar**, and **Pie** graphs so that you can easily read the data.

1. Log in to the Jenkins server.

Result

The Jenkins dashboard is displayed.

2. Open your Jenkins **Freestyle** or **Pipeline** project from the list.
3. Click **Build Now** to run the test assets from Jenkins.

Results

You have run the test from the Jenkins server.

What to do next

You can view the build logs by clicking the build number from the **Build History** pane, and then selecting the **Console Output** option. You can also view the history of test results in a tabular format by clicking the **Test Results Analyzer** option.

You can access the report URLs to view the test run information at any point in time. You can also view the test reports and logs of the test that was run on the Jenkins server from the **Results** page on Test Hub. See [Test results and reports on page 789](#).

Integration with Xray test management for Jira

When you use Jira as a change management system on HCL DevOps Test Hub (Test Hub) and you use Xray as the test management application in Jira, you can integrate Xray with Test Hub. With this integration, you can export the test results of the tests that run on Test Hub to your Jira project. You can then view and use the test results in your project in Jira.

Prerequisites

Before you integrate Xray with Test Hub, you must have completed the following tasks:

- Configured Jira as a change management system on Test Hub. See [Configuration of Atlassian Jira as a change management system on page 892](#).
- Installed the Xray test management app in Jira.



Note: The Xray integration is supported only when the Xray test management application is installed on a Jira cloud.

- Ensured that you obtained the access and authentication details to Jira from the Jira administrator.
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).

Task flow

You can follow the tasks that are listed in the following table when you want to export test results to your Xray test project in Jira. The table also provides you the links to the information about the tasks.

Task	Go to...
Generate the Xray API keys for your username on Jira.	Generating API keys for the Xray app in Jira on page 989
Use the generated Xray API keys to configure the Xray integration in Test Hub.	Configuring the Xray integration on page 990
Select an issue in your Xray project in Jira as a location to export the test results of a test run. You must select the issue while you configure a test run on Test Hub.	Selecting a Jira Xray issue to export test results on page 991

You can manage the Xray settings that you configured in Test Hub. You can edit the saved configuration or you can delete the configuration. For more information, see [Managing the Xray configuration on page 993](#).

Generating API keys for the Xray app in Jira

You must generate API keys for the Xray app in Jira and use them to configure the Xray integration in HCL DevOps Test Hub (Test Hub). The Xray API keys are used by Jira to authenticate the requests from Test Hub.

Before you begin

You must have completed the following tasks:

- Read about the prerequisite tasks in [Prerequisites on page 988](#).
- Verified that you can access Jira.
- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).

1. Log in to Jira.
2. Click **Settings > Apps**.

The apps that are installed in Jira are listed in the navigation pane.

3. Find the **Xray App** in the navigation pane, and then click the **API Keys** option.

The **API Keys** page is displayed.



4. Click **Create API Key**.

The **Create API Key** dialog is displayed.

5. Enter your username in the **User** field, and then select the username that is displayed.

6. Click **Generate**.

The API keys are generated and displayed under the **Client Id** and **Client Secret** fields.

7. Click the **Copy** icon  in the **Client Id** field to copy the key, and then paste the copied key into a text file.
8. Click the **Copy** icon  in the **Client Secret** field to copy the key, and then paste the copied key into a text file.

Results

You have generated the Xray API keys in Jira.

What to do next

You must use the Xray API keys to configure the Xray integration in Test Hub. See [Configuring the Xray integration on page 990](#).

Configuring the Xray integration

You can set up the Xray integration when you want to export test results from HCL DevOps Test Hub (Test Hub) to an issue in your Xray project in Jira. You must configure the Xray integration in your project in Test Hub before you select the Xray project in Jira as the location to export the test results after the test runs in Test Hub.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Generated the Xray API keys in Jira and saved the keys in a text file. See [Generating API keys for the Xray app in Jira on page 989](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Integration** in the navigation pane.

The **Integration** page is displayed.

4. Click the **TEST MANAGEMENT** tab, if it is not already open.

5. Click **Add Integration > Jira/Xray Software**.

The **Add Xray Integration** dialog is displayed.

6. Perform the following steps:

- a. Enter the Xray API key that was generated for the **Client Id** and copied into the text file, in the **Client Id** field.
- b. Enter the Xray API key that was generated for the **Client Secret** and copied into the text file, in the **Client Secret** field.
- c. Click **Save**.

The Xray configuration is saved and displayed on the **TEST MANAGEMENT** page.

Results

You configured the Xray integration on Test Hub.

What to do next

You can perform the following tasks:

- View the following information on the **TEST MANAGEMENT** tab that you used to configure the Xray integration:
 - The URL of the Jira server on which the Xray test management app is installed.
 - The project that you selected in Jira.
 - The Xray API key that is used as the Client ID.
 - The status of the Xray configuration.
- Select an issue in Jira as the location to which you can export the test results of a test run while you configure a run of the test on Test Hub. See [Selecting a Jira Xray issue to export test results on page 991](#).

Selecting a Jira Xray issue to export test results

After you configure the Xray integration on HCL DevOps Test Hub (Test Hub), you can select a Jira Xray issue in Jira or create a new issue in Jira to export the test results. The test results are exported to the selected issue after the tests run on Test Hub. You can then view and track the test results in your Jira project.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Configured the Xray integration by using the Xray API keys that are generated for your username. See [Configuring the Xray integration on page 990](#).
- Ensured that the **Execute test asset** dialog is displayed and you have configured the other settings for a test run in Test Hub.

About this task

When you configure a test run in Test Hub, you can select a Jira issue as the location where you want the test results to be exported. After the test is run, the results are exported to the Jira issue. You can then view and track the test results in the Jira issue from your project in Jira.


1. Click the **RESULTS** tab in the **Execute test asset** dialog.

The **RESULTS** tab is displayed.



Note: The **Export report options** field is enabled only if you configured the Xray integration with the Xray API keys that are generated for your username in Jira.

2. Perform the actions listed in the following table to select an option to export the test results:

Export options	Results in...	Actions
Jira Xray	The test results of the test that you configure for a run are exported to the selected issue after the test run is completed. The results are exported to the selected issue that exists in your Xray project in Jira.	Perform the following steps: <ol style="list-style-type: none"> a. Select the Jira Xray option. <p>The Select issue field is displayed.</p> <ol style="list-style-type: none"> b. Click the field to view the list of issues in the Xray project in Jira. Alternatively, you can enter the issue key in the Jira Xray project. c. Select the issue to which you want to export the test results. d. Continue with the next step.
New Jira item	The test results of the test that you configure for a run are exported to a new Jira issue. The issue is created in your project in Jira after the test run is completed.	Perform the following steps: <ol style="list-style-type: none"> a. Select the New Jira item option. b. Continue with the next step.
None	The test results are not exported.  Note: This is the default option.	Perform the following steps when: <ul style="list-style-type: none"> ◦ You do not want to export the test results. ◦ You selected a Jira issue in a previous run of the same test, and you do not want to export the results of the current run. <ol style="list-style-type: none"> a. Select the None option. b. Continue with the next step.

Results

You selected an issue in your Xray project or created an issue in Jira as a location to export the test results of a test run on Test Hub.

What to do next

After the test run is completed, you can go to the **Results** page and expand the test result to view the Xray issue key to which the test results were exported. You can then click the Xray issue key link to open the issue and view the details of the issue in Jira.

Managing the Xray configuration

You can manage the configuration of the Xray integration that you set in HCL DevOps Test Hub (Test Hub) when you want to modify the API keys configured or when you want to delete the existing Xray configuration.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Project Owner* or *Tester* in the project. See [Managing access to the server projects on page 1021](#).
- Ensured that you obtained the access and authentication details to Jira from the Jira administrator.
- Configured the Xray API keys for the Xray integration on Test Hub. See [Configuring the Xray integration on page 990](#).
- Regenerated the Xray API keys in Jira by following the steps to generate the Xray API keys in [Generating API keys for the Xray app in Jira on page 989](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.


The **Overview** page of the project is displayed.

3. Click **Manage > Integration** in the navigation pane.

The **Integration** page is displayed.

4. Click the **TEST MANAGEMENT** tab, if it is not already open.

The details about the Xray integration that is configured are displayed.

5. Click the **Open action menu** icon  of the Xray integration that you want to manage.

The **Update credentials** and **Delete configuration** options are displayed in the list.

6. Perform any of the following actions when you want to modify or delete the Xray configuration:
 - To modify the Xray API keys that were set previously in the Xray configuration, go to [Step 7 on page 993](#).
 - To delete the Xray configuration, go to [Step 8 on page 994](#).

7. Perform the following steps to modify the Xray API keys:

- a. Select **Update credentials** as the option from the list.
The **Update Xray Integration** dialog is displayed.
- b. Enter the Xray API key that was generated for the **Client Id** and copied into the text file, in the **Client Id** field.
- c. Enter the Xray API key that was generated for the **Client Secret** and copied into the text file, in the **Client Secret** field.
- d. Click **Update** to save the updated Xray API keys in the Xray configuration.

The Xray configuration is updated and a confirmation message is displayed.

8. Perform the following steps to delete the existing Xray configuration:

- a. Select **Delete** as the option from the list.
The **Delete Jira server configuration** dialog is displayed.
- b. Click **Delete**.

The Xray configuration is deleted and a confirmation message is displayed.



Note: The links to the issues that you created to export the test results are not deleted from the results panel in Test Hub even after you delete the Xray configuration.

Results

You achieved the following results:

- Updated the Xray configuration.
- Deleted the Xray configuration.

Managing access to DevOps Test Hub

Desktop clients and third-party integrations use offline user tokens to connect to HCL DevOps Test Hub (Test Hub).

You can configure the desktop clients and third-party integrations to access Test Hub by using the offline user tokens created from the server.

Offline user tokens are used in the following cases:

- To retrieve secrets configured on the server to be used in certain tests in HCL DevOps Test Integrations and APIs (Test Integrations and APIs).
- To enable publishing of test results and reports from desktop products that include Test Integrations and APIs, HCL DevOps Test Performance (Test Performance), or HCL DevOps Test UI (Test UI) to Test Hub.

- To enable the Resource Monitoring Service in Test Performance to monitor a data source during a run on Test Hub.
- To enable agents to connect with Test Hub.
- To enable HTTP proxies to connect with Test Hub.
- To enable third-party integrations.


About this task

From Test Hub, you can create an offline user token, copy it, and then use that token to enable connections to be authenticated by Test Hub.

You can also delete all the tokens, when you no longer need access or you want to prevent access to the server.

Creating a token

- Follow these steps:

1. Click the **User** icon  from the menu bar, and then select **Create Token**.

The Create Offline User Token dialog is displayed.



Note: This token is not accessible after you copy it as the dialog box closes.

2. Copy the token and paste it into a private location for ongoing reference.

The dialog closes.

3. Paste the token into the UI of the desktop product or in the application where this token is required.


See the following table for a list of where you can use the tokens and where you can find more information.

Table 53. Using the offline user tokens to access Test Hub

Use the offline user token in the following places	More information
Test Integrations and APIs	See .Test Integrations and APIs and Test Hub.
Test Performance	See .Publishing test results to the server. See Enablement of Resource Monitoring services for a schedule..
Test UI	See .Publishing test results to the server.
Third-party integrations	See Integrations with other applications on page 924.

Deleting tokens

- Follow these steps:

1. Click the **User** icon  from the menu bar and select **Delete Tokens**.

The **Delete All Offline User Tokens** dialog is displayed.

2. Inform the desktop client users that the tokens were deleted. Fix any test automation scripts that used tokens.

Working with DevOps Test Hub REST services

When you want to use the REST APIs of HCL DevOps Test Hub (Test Hub) in your test automation scripts, you can view the REST APIs that are exposed by Test Hub. You can also run the APIs from within your project in Test Hub.

All REST services are accessible from the base URL of Test Hub and are defined by using Swagger. A navigable user interface lists the available set of APIs at `<Test_Hub_URL>/#/Initial/Admin/docs`. This Swagger file describes the individual APIs along with details of the parameters and response bodies.

You must be assigned the role of an administrator to be able to view or run the REST APIs.

After you log in to Test Hub with your credentials that has administrator privileges, you must go to the following URL:

`<Test_Hub_URL>/#/Initial/Admin/docs`

The Swagger docs that are available can be viewed from the drop down list.

You can select the Swagger documentation for the REST APIs that you want to view or use. For example, you can select the `/jobservicev2/api-docs` to view the APIs for the execution micro-service.

You can perform the following steps to view or run the API:

1. Expand the section to view the APIs that are available.
2. Expand each of the APIs to view the details such as the description, examples, parameters, and responses.
3. Click **Try it out**.
4. Enter the values for the required parameters.
5. Click **Execute** to run the API.

You can view the response after the API runs.

You can view the API doc for the execution micro-service that is extracted in this [PDF](#) document.

Chapter 7. Test Manager Guide

This guide describes how to manage and track your overall test effort. This guide is intended for test managers.

Related information

[Managing access to the server projects on page 1021](#)

Managing team spaces

You might want to become a team space member, add members to your team space, change a user role in a team space, modify the team space, or delete a team space.

Each team space is associated with at least one *Team Space Owner*. You can assign multiple owners to your team space. Any user who becomes a member of a team space and can be assigned the role of a *Member*, *Project Creator*, *Architect*, or *Team Space Owner* of a team space, can access the team space.

As an owner of a team space, after you create a team space, you must configure the team space license. You can then add members to the team space and can assign different roles to each member of the team space. You can update or delete a team space. You can also add a repository to a team space to store a system model and edit the repository settings. Apart from adding a repository, you can add, update, or delete the Resource Monitoring agent if no longer required.

As an *Architect* of a team space, you can manage a team space repository and design a system model.

As a *Project Creator* of a team space, you can create a project in the team space.

As a licensed user who is assigned with a *Member*, *Project Creator*, *Architect*, or *Team Space Owner* role of a team space, you can add a Docker host to your team space. You can view the configured Docker, agents, or intercepts of your team space. You can edit only those Docker hosts that you created.

Viewing team spaces

After you log in to HCL DevOps Test Hub (Test Hub), you can view all the existing team spaces on the **Team Space Dashboard** page.

About this task

As a member of a team space, you can view your team spaces under **My Team Spaces**. You can also view the team spaces that are created by other users as the public team space and the Permissive team spaces under **Other Team Spaces**.

If the initial team space does not exist, then you can view the **Team Space Dashboard** without any navigation pane.

You as a member of a team space, when you want to use that team space immediately after you log in to Test Hub, you can suffix the alias in the application URL.


For example, if the alias of the team space is `first`, then you must use the following Test Hub URL in the address bar:

```
<server url>/#/first
```

1. Log in to Test Hub.

Result

The **Projects** page of the initial team space is displayed.

2. Click the **Settings** icon  from the left navigation pane, and then click **Join another team space**.

Result

The **Team Space Dashboard** is displayed.

You can view the team spaces in the following panels:

- **My Team Spaces:** You can view all the team spaces that you created. You can also view the team spaces in which you are a member.
- **Other Team Spaces:** You can view all the team spaces that are created by others.



Note: You cannot view the Restrictive private type team spaces.

Results

You have viewed the team spaces that are created by you and other users on the **Team Space Dashboard**.

What to do next

You can select a team space and can send a request to become a member of that team space. See [Becoming a team space member on page 1005](#).

Adding members to a team space

When you create a team space to manage projects on HCL DevOps Test Hub (Test Hub), you must add users to your team space so that they can create and access the projects.

Before you begin

- As an administrator, you are assigned the default role of a *Team Space Owner*. If you log in to Test Hub as another user, then you must have been assigned the role of a *Team Space Owner*.
- As an owner of a team space, you must have completed the following tasks:
 - Created a team space. See [Creating a team space on page 171](#).
 - Configured a license for the team space. See [Configuring licenses for team spaces on page 181](#).

You must have been assigned a role as a *Team Space Owner*. If you are a server administrator, then you are assigned with the default role of a *Team Space Owner*.

About this task

As an owner of a team space, you can add multiple users to your **Permissive** or **Restrictive** team space.

1. Log in to Test Hub.

Result

The **Projects** page of the initial team space is displayed.

2. Click the **Settings** icon  from the left navigation pane.



Note: If the initial team space does not exist, then you can view the **Team Space Dashboard** page without any navigation pane.

3. Identify the team space for which you want to add a member.

You can either use the same team space where you choose to work or switch to any other team space.

4. Click **Add member** to add members in the initial team space.

Alternatively, you can select the team space, and then click **Settings > Add member**.

Result

The **Add Member** page is displayed.

5. Enter a username in the **Add People** field to add users.

Alternatively, you can find the username by using type search.

Result

The list of users is displayed.

6. Select the user.
7. Assign a role to the selected user.




Note: All members are assigned the default roles of a *Project Creator* and *Architect*.

You can choose the role for the user based on its associated actions:

Roles	Responsibilities
<i>Team Space Owner</i>	<ul style="list-style-type: none"> ◦ Create a team space. ◦ Update and delete a team space. ◦ Add members to a team space. ◦ Remove members from the team space. ◦ Assign the roles to the members. ◦ Modify the roles of the members. ◦ Create and modify a project. ◦ Add or update the system model. ◦ Add a repository to store a system model. ◦ Update and delete a repository in a team space. ◦ Add, update, or delete Docker. ◦ Configure or update agent and intercepts. ◦ Register, update, delete, and list Resource Monitoring sources in a team space.

Roles	Responsibilities
<i>Project Creator</i>	<ul style="list-style-type: none"> ◦ Create, update, and view a project ◦ Add, update, or delete Docker. ◦ Configure or update agent and intercepts.
<i>Architect</i>	<ul style="list-style-type: none"> ◦ Create, update, delete, and view a repository in a team space. ◦ Create, update, and delete components in the system model. ◦ Add, update, or delete Docker. ◦ Configure or update agent and intercepts.
<i>Member</i>	<ul style="list-style-type: none"> ◦ View the list of team spaces in the Team Space Dashboard. ◦ View the configuration of the team space. ◦ Add, update, or delete Docker. ◦ Configure or update agent and intercepts. ◦ View the registered Dockers, agents, or intercepts. ◦ View the Resource Monitoring agents in a team space. ◦ View the registered Resource Monitoring agents. ◦ View components in the system model.

**Notes:**

- The members with the assigned role of a *Team Space Owner*, *Project Creator*, *Architect*, or *Member* of a team space can be assigned with any of the following project roles:
 - *Project Owner*
 - *Tester*
 - *Viewer*
- You can add or remove the role of a member by using the **Menu** icon  next to each user in the **Members** list.

8. Repeat the applicable steps to add another user.

Results

You have added members with assigned roles to your team space. You can view the list of all added users on the **Team Space Configuration** page.

What to do next

You can reassign or remove the roles of users to one or more team spaces that you own. See [Managing members and their roles in a team space on page 1006](#).

All users can request to be a member of another team space. See [Becoming a team space member on page 1005](#).

Users added as members to the team space can view the team space in **My Team Spaces**.

Modifying the configuration of a team space

After you create a team space on HCL DevOps Test Hub (Test Hub), if you want to modify the details of the team space, add a repository to your team space, add or remove members of a team space or delete a team space, then you can perform these tasks from the **Team Space Configuration** page.

Before you begin

You must have been assigned the role of a *Team Space Owner*.

About this task

The **Team Space Configuration** page displays details of the team space.

As an owner of a team space, you can modify details of the team space, add members to your team space, and assign roles to the members of the team space on the **Team Space Configuration** page.

The *Team Space Owner* and *Architect* can add or remove a repository from the team space.

If the team space is no longer required, then the *Team Space Owner* can also delete that team space.


As a *Member*, *Project Creator*, or *Architect*, you can only view the configuration details of a team space on the **Details** tab of the **Team Space Configuration** page.

As a *Member*, and *Project Creator* you can only view the configuration details of a team space on the **Repository** tab of the **Team Space Configuration** page.

1. Log in to Test Hub.

Result

The **Projects** page of the initial team space is displayed.



2. Click the **Settings** icon  from the left navigation pane.
3. Identify the team space for which you want to modify the configuration.
You can either use the same team space where you are working or switch to any other team space.
4. Click **Manage team space**.

Result

The **Details** page is displayed.

5. Perform the modifications on the following details of the **Details** tabs of the team space:

Fields	More information
Name	Displays the name of the team space.
URL alias	Displays an alias of the team space. You must suffix this alias in the application URL to access a specific team space after you log in to the application.
Available at	Displays the application URL by using an alias. You can copy this URL and use it when you want to ac-

Fields	More information
	<p>cess the team space immediately after you log in to the application.</p> <p> Note: You cannot modify the URL.</p>
Member Access	<p>Displays the type of team space that defines the permissions to access the team space. A team space can be of the following types:</p> <ul style="list-style-type: none"> ◦ Permissive: All licensed users can access this team space without the approvals of a <i>Team Space Owner</i>. ◦ Restrictive: All licensed users must get approvals from the <i>Team Space Owner</i> to access this team space.
Visibility	<p>The visibility mode of a team space can be of the following types:</p> <ul style="list-style-type: none"> ◦ Public: The team spaces are visible to all licensed users in the Other Team Spaces panel. ◦ Private: The team spaces are not visible to all the licensed users in the Other Team Spaces panel. <p> Note: The visibility mode is available only when the Restrictive type of team space is selected.</p>
Description	Brief description of the team space. This is an optional field.
Add People	List of members in a team space.

6. Click the **Repository** tab to modify the details of the repository of the team space.

Results

You have modified the configuration details of the selected team space.

What to do next

You can add members and assign roles to each member of your team space.

Related information

[Adding members to a team space on page 998](#)

[Managing members and their roles in a team space on page 1006](#)

[Adding repositories to a server project on page 1014](#)

Viewing the configuration of a team space

You can view the configuration of a team space as a member of any team space on HCL DevOps Test Hub (Test Hub). You can view the configuration details on the **Team Space Configuration** page.


Before you begin

You must be a member of a team space.

About this task

The **Team Space Configuration** page displays the configuration details of the team space.

As a member of a team space, if you want to access your team space immediately after logging into Test Hub, then you must suffix the alias in the application URL. You can find details of the alias on the **Team Space Configuration** page. You can also view the list of members and their roles in that team space on the **Team Space Configuration** page.


1. Log in to Test Hub.
2. Click the **Settings** icon  from the left navigation pane.
3. Identify the team space for which you want to view the configuration.
You can either use the same team space where you are working or switch to any other team space.
4. Click **Manage team space**.


Result

The **Details** page is displayed.

You can view the following details about the team space:

Fields	Tabs	More information
Name	Details	Displays the name of the team space.
URL alias	Details	Displays an alias of the team space. You must suffix this alias in the application URL to access a specific team space after you log in to the application.

Fields	Tabs	More information
Available at	Details	Displays the application URL by using an alias. You can copy this URL and use it when you want to access the team space immediately after you log in to the application.
Member Access	Details	<p>Displays the type of team space that defines the permissions to access the team space. A team space can be of the following types:</p> <ul style="list-style-type: none"> ◦ Permissive: All licensed users can access this team space without the approvals of a <i>Team Space Owner</i>. ◦ Restrictive: All licensed users must get approvals from the <i>Team Space Owner</i> to access this team space.
Visibility	Details	<p>The visibility mode of a team space can be of the following types:</p> <ul style="list-style-type: none"> ◦ Public: The team spaces are visible to all licensed users in the Other Team Spaces panel. ◦ Private: The team spaces are not visible to all the licensed users in the Other Team Spaces panel. <p> Note: The visibility mode is available only when the</p>

Fields	Tabs	More information
		 Restrictive type of team space is selected.
Members	Details	Displays a list of members of the team space.
Repository	Repository	Displays the repository details that host the system model.

Results

You have viewed the configuration details of the selected team space.

Becoming a team space member

After logging into HCL DevOps Test Hub (Test Hub), the licensed users must be a member of any team space to perform test activities in a project. As a licensed user, if you are not assigned the role of a member in any team space, then you can request the *Team Space Owner* to become a member of any existing team space.

Before you begin

You must be a licensed user.

About this task

If you want to become a member of a **Restrictive** team space, then only the owner of that team space can grant you the role of a member of a team space. If you request to become a member of a **Permissive** team space, then you automatically become a member of that team space.



Note: The server administrator can join any team space automatically without any approval and is assigned with a default role of *Team Space Owner*.

1. Log in to Test Hub.

Result

The **Projects** page of the initial team space is displayed.

2. Click the **Settings** icon  from the left navigation pane, and then **Join another team space**.

Result

The **Team Space Dashboard** is displayed.


You can view the team spaces in the following panels:

- **My Team Spaces:** You can view all the team spaces that you created. You can also view the team spaces in which you are a member.
- **Other Team Spaces:** You can view all the team spaces that are created by others.



Note:

You can view only the public type of **Restrictive** team spaces and the **Permissive** team spaces. You cannot view the **Restrictive** private type team spaces.

3. Select the team space that you want to join from the list of other team spaces in the **Team Space Dashboard**.
4. Click the **Key** icon  of the selected team space.
5. Confirm your request when prompted.

If you agree, then the *Team Space Owner* sees a notification about the request.

After opening the notification, the *Team Space Owner* can view the user who is requesting the access and can accept or decline the request. If the *Team Space Owner* accepts your request, you are added and assigned a *Member* role.



Notes:

- You can become a member of any team space only when the license of the selected team space is configured.
- If you are a server administrator, then the *Team Space Owner* role is the default role assigned.
- If you are a licensed user, then the *Member* role is the default role assigned and is restricted to specific actions.

You can then see the team space under **My Team Spaces** on the **Team Space Dashboard** page.

You might want to follow up with the *Team Space Owner* if your request is pending for some time.

Results

You have become a member of another team space.

What to do next

You can view the actions that you can perform based on your assigned role. See [Adding members to a team space on page 998](#).

Managing members and their roles in a team space

After you, as a *Team Space Owner* assign roles to the members of a team space on HCL DevOps Test Hub (Test Hub), you might want to remove a member from the team space or reassign the role to the existing members of your team space.

Before you begin

You must have been assigned the role of a *Team Space Owner*.

About this task


As a licensed user, you are assigned a role of a *Member* in any team space after the approval of the *Team Space Owner*.

As an owner of a team space, you can remove or reassign any member of your team space with the roles of a *Team Space Owner*, *Project Creator*, or *Architect*. You can also remove any member from the list of members of the team space.

1. Log in to Test Hub.

Result

The **Projects** page of the initial team space is displayed.

2. Click the **Settings** icon  from the left navigation pane.
3. Identify the team space for which you want to reassign the roles of the members or remove a member from the list of members.


You can either use the same team space where you are working or switch to any other team space.

4. Click **Add member** if you want to reassign the roles of the members or remove a member from the list of members of the initial team space.

Alternatively, you can select the team space, and then click **Settings > Add member**.

Result

The **Add member** page is displayed.

5. Identify the member from the list of **Members**.
6. Click the **Menu** icon  next to the selected member.
7. Perform any one of the following operations to the identified member of the team space:

Choose from:

- Click **Remove Member** to remove the identified member from the team space.
- Select the role that you want to assign or remove the assigned role for the identified member.

You can choose multiple roles for the members based on the associated actions:

Roles	Actions
<i>Team Space Owner</i>	<ul style="list-style-type: none"> ▪ Create a team space. ▪ Update and delete a team space. ▪ Add members to a team space. ▪ Remove members from the team space. ▪ Assign the roles to the members. ▪ Modify the roles of the members. ▪ Create and modify a project. ▪ Add or update the system model. ▪ Add a repository to store a system model. ▪ Update and delete a repository in a team space. ▪ Add, update, or delete Docker.

Roles	Actions
	<ul style="list-style-type: none"> ▪ Configure or update agent and intercepts. ▪ Register, update, delete, and list Resource Monitoring sources in a team space.
<i>Project Creator</i>	<ul style="list-style-type: none"> ▪ Create, update, and view a project. ▪ Add, update, or delete Docker. ▪ Configure or update agent and intercepts.
<i>Architect</i>	<ul style="list-style-type: none"> ▪ Create, update, delete, and view a repository in a team space. ▪ Create, update, and delete components in the system model. ▪ Add, update, or delete Docker. ▪ Configure or update agent and intercepts.
<i>Member</i>	<ul style="list-style-type: none"> ▪ View the list of team spaces in the Team Space Dashboard. ▪ View the configuration of the team space. ▪ Add, update, or delete Docker. ▪ Configure or update agent and intercepts. ▪ View the registered Dockers, agents, or intercepts. ▪ View the Resource Monitoring agents in a team space. ▪ View the registered Resource Monitoring agents. ▪ View components in the system model.



Note: The *Team Space Owner*, *Project Creator*, *Architect*, or *Member* of a team space can be assigned with any of the following project roles:

- *Project Owner*
- *Tester*
- *Viewer*

Similarly, you can remove the assigned role of any member of your team space.

Results

You have completed any of the following tasks in your team space:

- Reassigned the role of the members.
- Removed the assigned role of the members.
- Removed the members.

Related information

[Adding members to a team space on page 998](#)

Deleting a team space

You might want to delete a team space on when it is no longer needed.

Before you begin

You must have been assigned the role of a *Team Space Owner* of HCL DevOps Test Hub (Test Hub).

About this task

As a *Team Space Owner*, you can delete a team space that you do not need. When you delete a team space, it is a permanent deletion. You cannot undo the delete action on a team space.


If you delete the initial team space, then you can only view the **Team Space Dashboard** page without any navigation pane.

After you delete the team space, you can use the name of the team space again.

1. Log in to Test Hub.

Result

The **Projects** page of the initial team space is displayed.

2. Click the **Settings** icon  from the left navigation pane.
3. Identify the team space that you want to delete.

You can either use the same team space where you are working or switch to any other team space.

4. Click **Manage team space**.

Result

The **Details** page is displayed.

5. Click **Delete Team Space**.
6. Confirm that you want to delete the team space when prompted.

If you proceed, the team space is deleted including all of its projects and members.

Results

You have deleted the selected team space.

What to do next

You can view the available team spaces on the **Team Space Dashboard** page. See [Viewing team spaces on page 997](#).

After you delete a team space, the license that is used for that team space is released and is available to configure the other team spaces, if required. See [Configuring licenses for team spaces on page 181](#).

Managing DevOps Test Hub projects

HCL DevOps Test Hub (Test Hub) projects manage access to your test assets, which are stored in a Git repository. Projects are maintained in a team space. Projects are either public by default or private. Private projects are not discoverable by other users. You can either add your own project or you can request to be a member of another public project.

All the licensed users are assigned with a default role of a *Project Creator*. As a *Project Creator*, when you create a project immediately after logging in the application, the project is created in the initial team space. If you want to create a project in any specific team space, then you must become a member of that team space. When you add your own project, you can configure it now or later. Configuring the project includes adding details about the project, adding one or more Git repositories, optionally adding secrets, classifying encrypted datasets, and adding a change management system. To run test assets that are associated with an encrypted dataset, you must categorize the encrypted datasets by creating a classification. You can also add users to your project to access your test assets only if the users are the members of that team space. To use Jira to create defects for the tests available in your project, you must configure the Jira server for your project on Test Hub.

Related information

[Team space overview on page 168](#)

[Becoming a team space member on page 1005](#)

[Becoming a project member on page 1020](#)

[Managing access to the server projects on page 1021](#)

[Managing repositories on page 1026](#)

[Protecting API test assets by using secrets on page 1035](#)

[Managing an encrypted dataset on page 243](#)

[Default user administration on page 139](#)

[Configuring Jira as a change management system on page 893](#)

Working with projects in a team space

After you log in to HCL DevOps Test Hub (Test Hub), you must have a project to perform the test activities. To work with projects, you must create and configure projects.

Before you begin

You must have logged into Test Hub.

About this task

You must be assigned with the role of an *Owner*, *Tester*, or *Viewer* of a project.

1. Go to the **Projects** page from the **Initial Team Space** Home page. See [Viewing active projects on page 1018](#).
2. Add a project. See [Adding a project on page 1013](#).
3. Add a repository. See [Adding repositories to a server project on page 1014](#).
4. Add secrets in a classification. See [Secrets configuration on page 1017](#).
5. Encrypt the data set resources. See [Managing an encrypted dataset on page 243](#).
6. Configure a change management system. See [Configuration of a change management system on page 891](#).
7. Add members to a project and configure their roles. See [Adding users to a project on page 1017](#).
8. Select a global branch in a project to view the test assets or resources maintained in the repositories. See [Selecting the global branch in a project on page 1033](#)
9. Perform the following tasks based on the type of test assets or resources that you want to run:
 - a. Add agents to your project, if you are using remote agents as a location to run certain tests. See [Adding a performance agent to a project on page 389](#).
 - b. Add remote Dockers, if you are using a remote Docker host as a location to run tests. See [Adding a remote Docker host to the project for running tests on page 411](#).
 - c. Read the considerations that you must take into account before you configure a run or start a virtual service. See [Prerequisites for running HTTP virtual services on page 742](#).
 - d. Read the considerations that you must take into account before you configure a test run. See [Prerequisites to running tests on page 371](#).
10. Configure a run of the test asset or resource. See [Test run configurations on page 446](#).
11. View the progress of a test run. See [Viewing the progress of running test assets on page 731](#).
12. Manage a running test. See [Management of running tests on page 727](#).
13. View the results of a test run after the run is completed. See [Test results on page 788](#).
14. Review the test results and create defects for test runs from the Results page. See [Creating Jira issues for the test results on page 895](#).

Results

You have performed all the tasks with the projects.

Repository considerations for a server project

To collaborate with other project stakeholders, you can open test assets from a local clone of a remote repository, pull project test assets from the remote repository, and push changes made to your local test assets to the remote repository. Before you add a project and then add a repository to that project, you must consider some information about repositories.

Consider the following sections about version control systems that are used to manage remote repositories.

The following types of Git-based version control systems are supported:

- GitHub, which is an open-source product for storing and collaborating code.
- Azure Repos, which is part of a solution of Azure DevOps, and focuses on collaboration between DevOps professionals to develop private code.

Both these version control systems support Git repositories to store code.

You must be familiar with the various aspects of these version control systems to manage the application code that you want to test in HCL DevOps Test Hub (Test Hub).

You can learn more about installing, setting up, and using Git from the [Git documentation](#).

You can learn more about installing, setting up, and using Azure Repos from the [Azure Repos documentation](#).



Note: Git is the only tool that is supported to manage a repository in Azure Repos. Other tools, which can be used in Azure Repos, are not supported.

Git

You must install Git or upgrade to the supported version if you already have installed Git. See [System Requirements for DevOps Test Hub 2024.03 \(11.0.1\) on page 79](#).

Repositories and user identities

After you install Git, you must set up your Git repository and set up access for members. You must ensure that the repository contains your test assets.

Optionally, you can use a command line utility or Git tool to access the repository, upload your test assets, fetch or pull from the repository, push to the repository, clone the repository, and other operations that you want to perform in Git.

Local and shared repositories

After you create a remote or shared repository in Git, you can create a local version of the repository by cloning the remote repository. You must ensure that your test assets are available in the remote repository and are also cloned to the local repository.

Alternatively, if your test assets are on your local system, you can set up a Git repository in the bare mode, add the project files to the local repository, and then commit and push from the local repository to the remote repository in Git by using your preferred method.



Note: While copying the test assets from your local system to the repository, you must ensure that you copy the entire project that contains the test assets.

User authentication for the Git repository

The administrator can set up different types of authentications for accessing the Git repository. Test Hub supports the following authentication types:

- HTTP with username and password
- HTTP without username and password
- HTTPS with username and password

- HTTPS without username and password
- SSH with SSH key and passphrase
- SSH with SSH key and without a passphrase

Based on the authentication type that is set for a repository, you must provide the same authentication values in Test Hub when you add a repository.

Test assets

You must complete the following tasks in the desktop products where you author your test before you check in and commit the test assets to the Git repository.

Test type	Task	More information
API suite in Test Integrations and APIs	Change the local stub to a remote stub.	See Test run considerations for API Suites or API tests on page 373 .
	Add the library files.	

Adding a project

The first step is to add a project and provide some details about it.

Before you begin

To add one or more projects to manage access to your test assets, you must log in to HCL DevOps Test Hub (Test Hub) by providing the application URL in a browser.

If you are a new user, and LDAP and Active Directory are not configured, then you must first sign up by completing a form that specifies user information such as an email, user name, and password. You can then log in by using that information.

You must be a member of any licensed team space.

About this task

As a licensed user, you are by default assigned as both *Project Creator* and *Architect* role for the initial team space. You must select a team space to create a project.



You can view all the team spaces in the **Team Space Dashboard**. To create a project in any team space, you must be a member of that team space and assigned a *Team Space Owner* or *Project Creator* role in that team space.

After you select a team space from the **Team Space Dashboard**, you can add a project and give it a name and description.

1. Log in to Test Hub.

Result

The **Projects** page of the initial team space is displayed.

2. Click **New project** or  icon from the navigation pane to create a project in the team space where you are working.
Alternatively, if you want to create a project in any other team space, then switch to another team space from the **Settings** icon  in the navigation pane, and then create a project.

Result

The **Details** page is displayed.

Results

You have successfully created a project.

What to do next

You can modify your project configuration. You can also add a repository to your project.

Related information

[Default user administration on page 139](#)

[Changing the visibility configuration of a project on page 1019](#)

[Adding repositories to a server project on page 1014](#)

Adding repositories to a server project

You can add a single repository individually or multiple repositories simultaneously to a server project to access assets and resources that are available in the repositories.

Before you begin

You must have completed the following tasks:

- Added a project on HCL DevOps Test Hub (Test Hub).
- Been granted permission and you must have set up your credentials to access the remote repositories in a repository domain.
- Created test assets and resources that are contained in either a single git repository or multiple git repositories.

About this task

As a project member with the role of a *Project Owner* or a *Tester*, you can add a single repository or multiple repositories to your project.



Note: You can only add a single repository that is managed by using Git in Azure Repos. The addition of multiple repositories in Azure Repos is not supported.

You can use the information in the following table to decide the method for adding repositories to your project:

When...	Action...
<p>You want to add any of the following repositories:</p> <ul style="list-style-type: none"> • A single repository owned by you. • A single repository owned by other people. • The repositories that are in different domains. 	<p>You can add individual repositories one at a time.</p> <p>You must then provide the complete URL of the repository along with the required credentials to authenticate the connection to the repository from Test Hub.</p> <p>For example, the URL to a single repository can be as <code>https://github.com/<user>/<repository>.git</code>. You must provide the <i>username</i> and <i>password</i> to access this repository.</p>
<p>You want to add any of the following repositories:</p> <ul style="list-style-type: none"> • Multiple repositories that are owned by you and are contained in the same domain. • Multiple repositories that are owned by others and are contained in the same domain. 	<p>You can add all the repositories simultaneously that are in the same domain.</p> <p>You must then provide the URL of the repository domain along with the required credentials to authenticate the connection to the repository from Test Hub.</p> <p>For example, the URL of the repository domain can be <code>https://github.com</code>. You must select the platform and provide your <i>personal access token</i> to access the repositories.</p>

1. Log in to Test Hub and open the team space that contains your project.
2. Open your project either from the navigation pane or from the **My Projects** page.

The **Overview** page of the project is displayed.

3. Click **Manage > Configuration** from the navigation pane.
4. Click the **REPOSITORIES** tab.
5. Click **Add repository**.

The **Add repository** dialog is displayed.

You can either choose to add a single repository individually or multiple repositories simultaneously from a single domain.

6. Perform either of the following actions:
 - To add a single repository, go to [Step 7 on page 1015](#).
 - To add multiple repositories from a domain, go to [Step 8 on page 1016](#).
7. Perform the following steps to add individual repositories at a time:

- a. Enter the complete URL of the repository in the **Git URL** field.

For example, the URL to a single repository can be as `https://github.com/<user>/<repository>.git`.

The required authentication fields are displayed that depend on the protocol you specified in the URL.

If...	Then...
The repository URL contains <i>http</i> or <i>https</i>	The following fields for authentication are displayed: <ul style="list-style-type: none"> ▪ Username ▪ Password
The repository URL contains <i>git@</i>	The following fields for authentication are displayed: <ul style="list-style-type: none"> ▪ SSH/Deploy Key ▪ Passphrase



Note: If you use SSH to connect to a remote repository and Test Hub displays an `Auth Fail` error, you can resolve this error by regenerating your SSH keys by using the **-m PEM** option.

- b. Enter the credentials to authenticate the connection to the repository.
- c. Click **Add** to add the repository to your project.

You can add other individual repositories by specifying the URL and performing the actions in [Step 7 on page 1015](#).

8. Perform the following steps to add multiple repositories simultaneously that are contained in the same domain:



Note: You can only add a single repository that is managed by using Git in Azure Repos. The addition of multiple repositories in Azure Repos is not supported.

- a. Enter the URL of the repository domain in the **Git URL** field.
- For example, the URL of the repository domain can be `https://github.com`.
- b. Select the platform from the list under **Platform** where your repositories reside.
- c. Enter your *personal access token* to authenticate your connection to the repository domain.



Note: If you do not have your *personal access token*, you can click **Generate** to generate your *personal access token* and use the token to authenticate your connection.

d. Click **Next**.

The connection to the repository domain is established and all the repositories that you own are listed. The time when the repository was last updated is displayed in the **Last updated** column.

e. Select the repositories that you want to add to your project.

f. Click **Add**.



Note: Depending on the size of the repository that you are adding, it can take a few to several minutes to add the repository to your project on the server.

All the repositories that you selected are added to your project and are listed on the **Repositories** page.

Results

You have added individual repositories or multiple repositories to your project on the server.

What to do next

You can perform the following actions on the repository that you added:

- Update the authentication credentials if they are changed in the Git repository configuration. See [Updating the authentication credentials of the repository on page 1030](#).
- Delete a repository if it is no longer required. See [Deleting a repository on page 1031](#).
- Refresh a repository to fetch and synchronize changes from the remote repository. See [Refreshing repositories manually on page 1031](#).

Secrets configuration

If you are working with an API suite and the project test asset contains environment variables that are required for the test runs, you must configure the environment variables as secrets in a classification by using the **Add Secret** option in the **Security** page. Configuring secrets enable the API suite to run correctly on Test Hub.

Related information

[Protecting API test assets by using secrets on page 1035](#)

Adding users to a project

You can add users to your project in HCL DevOps Test Hub (Test Hub) so that they can access your test assets.

Before you begin

- You own a project.
- The users you want to add to your project must be the members of the team space of that project.


About this task

As a project owner, you can add other users to your public or private project. For example, if you have a large number of test assets that must be run, you might want your test team to help run them, and see the results.

1. Log in to Test Hub.

Result

The **Projects** page of the Initial Team Space is displayed.

2. Select the project for which you want to add the users.
3. Click the **Settings** icon  on the Project card.
4. Add users by entering the username of the user you want to add. You can add a partial name and then press the **Enter** key to see the user that you want to add.
5. Select that user and then assign a role.
6. Click **Viewer**, **Tester**, or **Owner**. When you add a user, the default role is *Viewer*.

The added user after logging in can see the owner's project in their list of **My Projects**.

7. Repeat the applicable steps to add another user.

Results

You have successfully added users to your project.

What to do next

You can add or remove roles to one or more projects that you own. All users can request to be a member of another public project.

Related information

[Becoming a project member on page 1020](#)

[Managing access to the server projects on page 1021](#)

Viewing active projects

After you log in to HCL DevOps Test Hub (Test Hub), you can view both active and archived projects in the **Initial team space**.

About this task

As a project owner, you can view your projects under the **My Projects** panel, and the projects that are created by other users as public projects are displayed under the **Other Projects** panel. You can view a list of all active projects that you created under **My Projects**. You can view a list of all archived projects under **Archived Projects**.

Log in to Test Hub.

The **Projects** page is displayed.

You can view the following projects:

- The active projects that you created or added to as a member with either the *Tester* or *Viewer* role are displayed under the **My Projects** panel.
- The active projects that are created by others as public projects are displayed under the **Other Projects** panel.

Results

You have viewed the active projects that are created by you and other users in the team space.

What to do next

You can perform any of the following tasks:

- Create a project. See [Adding details to a server project on page 1013](#).
- Request to become a member of a project that is displayed under the **Other Projects** panel. See [Becoming a project member on page 1020](#).
- View the other pages in the team space and perform tasks on those pages. See [Tasks in a team space on page 173](#).

Related information

[Archiving or unarchiving server projects on page 1023](#)

Changing the visibility configuration of a project

After you create a project in HCL DevOps Test Hub (Test Hub), you might want to change the visibility configuration of the project from the public, which is the default, to private.

Before you begin

You must have created a project.

1. Log in to Test Hub.

Result

The **Projects** page of the initial team space is displayed.

2. Select the project from the left navigation pane.

Result

The project page is displayed.

3. Click **Manage > Configuration >** .

Alternatively, select a project, and then click the **Configure project** icon  on the project card.

Result

The **Details** page is displayed.

4. Select **Private** as the visibility mode of the project.
5. Click **Save**.

Results

You have successfully changed the visibility configuration of a project.

What to do next

You can add a repository to your project.

Related information

[Default user administration on page 139](#)

Becoming a project member


You might want to request to be a member of another project. New users without any projects might also want to be a member of an existing project.

About this task



As a project owner, you can add users to your public or private projects. All users can request to become a member of a public project.




Note: You can add a licensed user as a member of your project only if that user is the member of your team space.

1. Request to be a member of another project. Search for the project that you want to join in the list of other projects from the project Home page. Click that Project card or the **Key** icon . Only public projects are visible in the list of projects.
2. Confirm your request when prompted. If you agree, the project owner is notified. The project owner sees a pending request **Notification** icon on their Project card.

My Projects

 myproj


No description.

Owner: [user55](#) 

Created: [a month ago](#)

Branch: **master**

Test Suites: **59** Individual Tests: **40**

Last Run Time: Nov 4, 2020 8:59 AM

from beginning

1	0/59	1/59
In Progress	Fail	Success

After opening the notification, the owner can see the user that is requesting access and can accept or decline it.

If the project owner accepts your request, you are added as a member of the project with a Viewer role. A Viewer role is the default role assigned and is restricted to specific actions.

You then see the project under My Projects on the Home page.

If the project owner declines your request, follow up with the project owner.

Results

You have successfully sent a request to become a member of another project.

Managing access to the server projects

You might want to add or remove a project member role from your project.

About this task

As a project owner, you can assign access by specifying a role when you add a user to your project. You can also assign more access or remove access for a user. Roles enable users to perform tasks on project resources such as running tests and viewing test results.

1. Choose the role in a project based on its associated actions.

Option	Description
Owner of a project	<ul style="list-style-type: none"> ◦ All Tester's actions ◦ Update, archive, list, and view a project

Option	Description
	<ul style="list-style-type: none"> ◦ Assign, update, delete, and list user roles for a project ◦ Accept new member requests ◦ Create, update, and delete a dataset classification in a project ◦ Add, update, and remove an encrypted dataset from a classification ◦ Update current row for a listed dataset in a project ◦ Delete, list and view a report ◦ Update, delete, and list a repository in a project ◦ Get the test assets from a repository in a project ◦ Create, update, delete and list secrets in a project ◦ Create, update, delete, and list resource monitoring sources in a project ◦ Get the content of a resource monitoring source in a project ◦ Register, update, delete, and list resource monitoring agents ◦ Create, configure, and delete the Jira change management system ◦ Create and update defects in the Jira change management system ◦ Create and update components in the system model
<p>Tester of a project</p>	<ul style="list-style-type: none"> ◦ All Viewer's actions ◦ List repositories in a project ◦ Run a test and create a report ◦ Stop a test while running ◦ Get the test assets from a repository in a project ◦ Get a list of datasets in a project ◦ Get the content of a dataset ◦ Update the current row for a listed dataset in a project ◦ Add, update, and remove an encrypted dataset from a classification

Option	Description
	<ul style="list-style-type: none"> ◦ Delete a report if you created the report ◦ Create and update defects in the Jira change management system ◦ Create and update components in the system model
Viewer of a project	<ul style="list-style-type: none"> ◦ List your own roles on a project ◦ List running tests ◦ List and view a report ◦ List resource monitoring sources in a project ◦ Get the content of a resource monitoring source in a project ◦ List the resource monitoring agents ◦ View components in the system model ◦ View defects in the Jira change management system
Non-member of a project	<ul style="list-style-type: none"> ◦ Request to be a member of a project

2. Open the project that you own.

From the project details, you can see a list of project members for your project and then add or remove a members role.

3. Add or remove the member access by using the **Menu** icon  next to each user in the member list.


Archiving or unarchiving server projects

You can archive a project when that project is no longer needed but you want to retain it for future reference. If you do want to use an archived project, you can unarchive it.

About this task

As a project owner, you can archive inactive projects in HCL DevOps Test Hub (Test Hub). Archived projects are not visible to users or project members, but owners can show or hide their archived projects. You can view the list of all archived projects from the navigation pane. Archived projects can be unarchived.

- To archive a project:

1. Log in to Test Hub.
2. Select your project from the **Projects** page.
3. Click the **Configure project** icon  on the project card.

Result

The **Details** page is displayed.

4. Click **Archive Project**.

5. Confirm that you want to archive the project when prompted.

If you proceed, the project is archived and is hidden from **My Projects**.

• To view an archived project:

1. Log in to Test Hub.

2. Select your team space.

3. Click **Archived Projects**.


Result

A list of all archived projects that are associated with you is displayed under **My Projects** and the archived projects that are associated with others are displayed under **Other Projects**.

• To unarchive a project:

1. Log in to Test Hub.

2. Click **Archived Projects**.

3. Select your archived project, and then click the **Configure project** icon  on the project card.

Result

The **Details** page is displayed.

4. Click **Unarchive Project**.

5. Confirm that you want to unarchive the project when prompted.

If you proceed, the project is unarchived and is shown under **My Projects**.

Deleting server projects

You might want to delete a project when the project is no longer needed, and you want to free up disk space.

About this task

As a project owner, you can delete a project that you no longer want. When you delete a project it is permanent. You cannot undo the delete project action.

After you delete the project, you can use the project name again.

1. Open the project that you want to delete.

2. From the project details, delete the project.

3. Confirm that you want to delete the project when prompted. If you proceed, the project is deleted including all of its test assets, results, and members.



Note: The repository that you used for the project is not deleted.

An overview of test assets, modifications, and scheduled runs

After you configure a project with one or more Git repositories and run some test assets, you can view statistics about those test assets.

From the **Overview** page, several cards display charts, sliders, and tables of the statistics of the test assets in your server project. You can click some of the charts, sliders, or tables to see more information, such as the progress of a scheduled run or the results of a run. You can also review some statistics by date and see more details such as the proportion of test assets by type.

You can view a server project, which might contain multiple repositories and multiple branches, by selecting the name of the branch from the **Branch** drop-down list. For more information about selecting branches, see [Selecting the global branch in a project on page 1033](#).

The **Members** area displays the total number of users who have access to the project. You can click the number to view the details of all the members. The **Owner** area displays the username of the project owner.

Test suites

This card displays a slider of the percentage of test suites by type. A test suite is a collection of tests, including schedules, Compound Tests, API Suites, and AFT Suites. You can hover over the slider and see the proportion of test suites by test suite type. The number of suites shown are relative to the selected branch.

Individual tests

This card displays a slider of the percentage of individual tests by type. You might have performances tests, API tests, or UI tests according to the desktop clients that you used to create the tests. You can hover over the slider and see the proportion of individual tests by type. The number of tests shown are relative to the selected branch.

Execution results

This card displays a chart of the proportion of test suites that were run by status and by date. If no date is selected, the card displays all the run results associated with the project since it was created. You can see the run results by clicking a test or Suite status. You can also hover over the chart and see the test or Suite status and the proportion of test assets. You can view statistics by tests or Suites that ran.



Notes:

- The results shown are generated from tests or Suites run.
- The statistics exclude canceled runs of tests or Suites.



- The number of Suites and individual tests that are associated with your project are shown on the **Home** page along with a verdict summary. You can see the run results by clicking a verdict. You can also see the last date and time when you ran one or more test assets.
- The number of suites shown are relative to the selected branch.

Last run

This card displays a table of the last three runs by date and time. You can see the run results by clicking each last run.

New or modified

This card displays a chart of new and modified test assets that were made in your Git repository for a week or for the last seven days based on a selected date range. You can hover your mouse over each bar in the chart and see the proportion of test assets by type. The data shown is relative to the selected branch.

Scheduled runs

This card displays a table of the next three scheduled runs by date and time. You can see the progress of a scheduled run by clicking each scheduled run.

Insights

This card displays a table of the insights on the supported parameters identified from the tests that were run in the project. The table lists the level of confidence with which the analysis of the specific parameter was done, the parameter analyzed, the name of the test asset, and the timestamp of the completed analysis. You can click any item in the row of a specific insight to view the details of that insight on the **Insights** page.

Related information

[Managing DevOps Test Hub projects on page 1010](#)

Managing repositories

After you create a project in HCL DevOps Test Hub (Test Hub), you can add repositories that contain test assets and resources to the project. You can also change the repository details, update the user credentials of a repository, delete a repository, create a webhook for the configured repository.

Tasks in a repository

To add a repository in your project and work on the repository, you can perform the tasks listed in the following table:

Task	More information
Adding a repository	Adding repositories to a server project on page 1014
Updating the repository credentials	Updating the authentication credentials of the repository on page 1030
Refreshing a repository	Refreshing repositories manually on page 1031
Creating a webhook for the repository	Creating webhooks on page 1034
Selecting the global branch	Selecting the global branch in a project on page 1033
Deleting a repository	Deleting a repository on page 1031

Adding repositories to a server project

You can add a single repository individually or multiple repositories simultaneously to a server project to access assets and resources that are available in the repositories.

Before you begin

You must have completed the following tasks:

- Added a project on HCL DevOps Test Hub (Test Hub).
- Been granted permission and you must have set up your credentials to access the remote repositories in a repository domain.
- Created test assets and resources that are contained in either a single git repository or multiple git repositories.

About this task

As a project member with the role of a *Project Owner* or a *Tester*, you can add a single repository or multiple repositories to your project.



Note: You can only add a single repository that is managed by using Git in Azure Repos. The addition of multiple repositories in Azure Repos is not supported.

You can use the information in the following table to decide the method for adding repositories to your project:

When...	Action...
<p>You want to add any of the following repositories:</p> <ul style="list-style-type: none"> • A single repository owned by you. • A single repository owned by other people. • The repositories that are in different domains. 	<p>You can add individual repositories one at a time.</p> <p>You must then provide the complete URL of the repository along with the required credentials to authenticate the connection to the repository from Test Hub.</p>

When...	Action...
	For example, the URL to a single repository can be as <code>https://github.com/<user>/<repository>.git</code> . You must provide the <i>username</i> and <i>password</i> to access this repository.
<p>You want to add any of the following repositories:</p> <ul style="list-style-type: none"> • Multiple repositories that are owned by you and are contained in the same domain. • Multiple repositories that are owned by others and are contained in the same domain. 	<p>You can add all the repositories simultaneously that are in the same domain.</p> <p>You must then provide the URL of the repository domain along with the required credentials to authenticate the connection to the repository from Test Hub.</p> <p>For example, the URL of the repository domain can be <code>https://github.com</code>. You must select the platform and provide your <i>personal access token</i> to access the repositories.</p>

1. Log in to Test Hub and open the team space that contains your project.
2. Open your project either from the navigation pane or from the **My Projects** page.

The **Overview** page of the project is displayed.

3. Click **Manage > Configuration** from the navigation pane.
4. Click the **REPOSITORIES** tab.
5. Click **Add repository**.

The **Add repository** dialog is displayed.

You can either choose to add a single repository individually or multiple repositories simultaneously from a single domain.

6. Perform either of the following actions:
 - To add a single repository, go to [Step 7 on page 1028](#).
 - To add multiple repositories from a domain, go to [Step 8 on page 1029](#).
7. Perform the following steps to add individual repositories at a time:
 - a. Enter the complete URL of the repository in the **Git URL** field.

For example, the URL to a single repository can be as `https://github.com/<user>/<repository>.git`.

The required authentication fields are displayed that depend on the protocol you specified in the URL.

If...	Then...
The repository URL contains <i>http</i> or <i>https</i>	The following fields for authentication are displayed: <ul style="list-style-type: none"> ▪ Username ▪ Password
The repository URL contains <i>git@</i>	The following fields for authentication are displayed: <ul style="list-style-type: none"> ▪ SSH/Deploy Key ▪ Passphrase



Note: If you use SSH to connect to a remote repository and Test Hub displays an `Auth Fail` error, you can resolve this error by regenerating your SSH keys by using the **-m PEM** option.

- b. Enter the credentials to authenticate the connection to the repository.
- c. Click **Add** to add the repository to your project.

You can add other individual repositories by specifying the URL and performing the actions in [Step 7 on page 1028](#).

8. Perform the following steps to add multiple repositories simultaneously that are contained in the same domain:



Note: You can only add a single repository that is managed by using Git in Azure Repos. The addition of multiple repositories in Azure Repos is not supported.

- a. Enter the URL of the repository domain in the **Git URL** field.
For example, the URL of the repository domain can be `https://github.com`.
- b. Select the platform from the list under **Platform** where your repositories reside.
- c. Enter your *personal access token* to authenticate your connection to the repository domain.



Note: If you do not have your *personal access token*, you can click **Generate** to generate your *personal access token* and use the token to authenticate your connection.

- d. Click **Next**.

The connection to the repository domain is established and all the repositories that you own are listed. The time when the repository was last updated is displayed in the **Last updated** column.

- e. Select the repositories that you want to add to your project.
- f. Click **Add**.



Note: Depending on the size of the repository that you are adding, it can take a few to several minutes to add the repository to your project on the server.

All the repositories that you selected are added to your project and are listed on the **Repositories** page.

Results

You have added individual repositories or multiple repositories to your project on the server.

What to do next

You can perform the following actions on the repository that you added:

- Update the authentication credentials if they are changed in the Git repository configuration. See [Updating the authentication credentials of the repository on page 1030](#).
- Delete a repository if it is no longer required. See [Deleting a repository on page 1031](#).
- Refresh a repository to fetch and synchronize changes from the remote repository. See [Refreshing repositories manually on page 1031](#).

Updating the authentication credentials of the repository

If the authentication credentials of the Git repository which was added to your project are changed, then you must update the credentials of the repository in HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Been assigned a *Member* or *Project Creator* role in a team space.
- Been assigned the *Tester* or *Owner* role in the project with access permissions to edit repositories.

About this task

In Test Hub, the repository is refreshed at regular intervals to pull the latest changes committed to the Git repository. After you update your authentication credentials of the repository, the repository is refreshed to ensure that you have access to the Git repository and to the latest assets in the repository.

1. Log in to Test Hub and open the team space that contains your project.
2. Open your project either from the left navigation pane or from the **My Projects** page.
3. Click **Manage** from the left navigation pane.

The **Repositories** page of the project is displayed.

4. Click the **Open action menu** icon inline with the repository that you want to update, and then click **Change...**

The **Edit Repository** dialog is displayed.

5. Enter the new credentials, and then click **Update**.

The new credentials are updated and your access to the repository is retained.

Deleting a repository

You can delete repositories that you no longer require in your test environment.

Before you begin

You must have completed the following tasks:

- Configured the repository that contains the test assets in your project on HCL DevOps Test Hub (Test Hub).
- Been assigned a *Owner* or a *Tester* role with access to delete repositories.

About this task

You can delete a repository that you have configured for your project.

1. Log in to Test Hub and open the team space that contains your project.
2. Open your project either from the left navigation pane or from the **My Projects** page.
3. Click **Manage** from the left navigation pane.

The **Repositories** page of the project is displayed.

4. Click the **Open action menu** icon inline with the repository that you want to update, and then click **Delete**.

The **Delete repository** dialog is displayed.

5. Select the checkbox, and then click the **Delete** button.

The repository is deleted and removed from the list.

Refreshing repositories manually

You can manually trigger the refresh activity for a repository to instantly fetch and synchronize the changes from the remote repository.

Before you begin

You must have completed the following tasks:


- Configured the repository that contains the test assets in your project on HCL DevOps Test Hub (Test Hub).
- Been assigned a *Member* or a *Project Creator* role.
- Been assigned a *Owner* or a *Tester* role.

About this task

Every two minutes, the repository is refreshed automatically. You can also manually trigger the refresh activity for a repository.

1. Log in to Test Hub and open the team space that contains your project.
2. Open your project either from the left navigation pane or from the **My Projects** page.
3. Click **Manage** from the left navigation pane.

The **Repositories** page of the project is displayed.





4. Click the **Open action menu** icon  inline with the repository, and then select **Refresh**.

The refresh activity is triggered.



Note: The **Refresh** option is disabled when the refresh is in progress. After the refresh is completed, the **Refresh** option is enabled.

The following table lists the icons and their description based on the status of the refresh activity:

Refresh icon	Description
	The last refresh activity was successful.
	<p>The last refresh activity was successful but errors were found while processing some of the test assets. To view the error details of the test assets for a specific repository, you can click the arrow inline with the repository name.</p> <p>You can resolve the errors of these test assets in the Git server, and then refresh the server repository, if required.</p> <p> Note: The repository, which contains the branch that is selected in the Branch field, displays the unprocessed test assets for that branch. Other repositories display the unprocessed test assets for their default branch.</p>
	<p>The refresh activity was not successful due to any one of the following reasons:</p> <ul style="list-style-type: none"> ◦ Internal error ◦ Connection issue ◦ Authentication failure ◦ Disk quota exceeded ◦ Repository unreachable

Results

You have manually refreshed the repository and the **Last Refreshed** field is updated with the following details:

- The time when it was last refreshed in the remote repository.
- Latest commit ID and message of the remote repository.

Selecting the global branch in a project

After you add one or more repositories to your project on HCL DevOps Test Hub (Test Hub), you can start accessing all branches in all repositories of that project. You can select only one branch as a global branch from the list of branches in the **Execution**, **Datasets**, and **Overview** pages.

Before you begin

You must have completed the following tasks:

- Been assigned a *Member* or *Project Creator* role in a team space.
- Added at least one repository in your project.

About this task

You can access all the branches contained in all the repositories, which are added to the project, by using the **Branch** field on the **Execution**, **Datasets**, and **Overview** pages.

After you select a branch from any of the repositories as the global branch, this selection is retained for you as the global branch unless you change the branch on any of the following pages: **Overview**, **Execution**, or **Datasets** pages.



Note:

- When you add the first repository to a project, the global branch is set to the default branch of the repository.
- Branches are arranged in an increasing order of their names in the list.
- When you change the branch in one of the pages, then the same branch is displayed as selected in the other pages and the contents of that page displays the assets from the selected branch.

1. Log in to Test Hub and open the team space that contains your project.
2. Open your project either from the left navigation pane or from the **My Projects** page.

The project **Overview** page is displayed.

3. Select a branch from the **Branch** field.

The branch that you select on any of the following pages is reflected in the rest of the other pages and the test assets are filtered based on the global branch selection on each of these pages.:

- **Overview**
- **Execution**
- **Datasets**

Results

You have selected a global branch in a project.


Creating webhooks

You can configure a webhook for a repository in the Git server to notify HCL DevOps Test Hub (Test Hub) whenever there is a push event in the remote repository. Then, the refresh activity is triggered to fetch the latest changes to the server repository.

Before you begin

You must have completed the following tasks:

- Configured the repository that contains the test assets in your project.
- Been assigned a *Member* or *Project Creator* role in a team space.
- Been assigned the *Tester* or *Owner* role in the project.

1. Log in to Test Hub and open the team space that contains your project.
2. Open your project.
3. Go to the **Repositories** page.
4. Click the **Menu** icon  and then click **Webhook**.

The **Webhook Configuration** dialog is displayed.

5. Copy the URL and the Secret of the repository that you want to configure.



Note: The URL and the Secret details are specific to a repository.

6. Click the link of the repository for which you want to configure the webhook.

The remote repository page is displayed.

7. Paste the details, which you copied from Test Hub, in the appropriate fields of the remote repository.

Results

You have configured the webhook for a repository in the Git server. The changes of the remote repository are synchronized in the repository of your project.

Protecting API test assets by using secrets

Secrets are key-value pairs that are created for your project in HCL DevOps Test Hub (Test Hub) under a classification. You can create classification for your project that enable you or members in your project to use secrets at test runtime either in Test Hub or in HCL DevOps Test Integrations and APIs (Test Integrations and APIs).

The secrets in a project in Test Hub maintains a separate access control list managed by the members with access to the secrets added to the classification. Controlling access to secrets means controlling access to applications and systems under test. The introduction of secrets under classification for a project simplifies managing access to separate environments. If a member of a project does not have access to a secret, for example, a *server credential* then the member cannot accidentally or intentionally run tests against that server. For example, tests that must access the database server by using the server credentials to retrieve stored data can only be run by a member if the access to the secrets is granted.



Note: Secrets are applicable to test assets created in Test Integrations and APIs that facilitates running tests in defined environments. Secrets are not applicable to tests created in Test UI or Test Performance.

As a project member with the *Owner* or *Tester* role, you can add secrets to the classification in the project. You can grant or restrict access to the secrets that you added to the classification in the project.

Members with access to a secret classification can access, edit, or delete the secrets in Test Hub and can view secrets, edit secrets, or delete secrets.

Members are granted Read access to the classification at the role level, while Read and Write access can be provided to specific members.

Members in the project with the *Owner* or *Tester* role and with access to the secrets can use the secrets in tests at runtime.

If you are configuring a project to run an API Suite with tests that refer to secret values, you must configure the secrets under a classification by using the **Add Secret** option in the **Security** page. You must complete the following tasks:

1. Create a classification. See [Creating a classification on page 193](#).
2. Add secrets in the classification. See [Creating a secret in a classification on page 1035](#).
3. Grant access to project members or member roles, who can access the secrets. See [Granting access to members or member roles on page 1037](#).

Creating a secret in a classification

You must create secrets in the classifications that is set in your project so that the members of the project with access to the secrets can use them in specific during an API Suite run.

Before you begin

You must have completed the following tasks:

- Created a project on Test Hub.
- Created a classification. See [Creating a classification on page 193](#).
- You must be a member with access to the secrets in the classification. See [Granting access to members or member roles on page 1037](#).

About this task

You can also configure secrets such that the secrets can be used across different test environments by members with access to the classification. Secrets correspond to the environment variables or tags that you create in a Test Integrations and APIs project specific to an environment.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

The **Security** page is displayed.

4. Click a classification for which you want to add the secrets under the **Classification**.

5. Click **Add Secret**.

Result

The **Add Secret** dialog is displayed.

6. Enter a name for the secret as its *Identifier*, and then enter the `password` as the *Value* for the secret.

For example, enter the name of the secret to access a database as `dbcred` and enter the `password` required to access the database as its value.

7. Click **Create**.

A message is displayed that confirms the creation of the secret.

Results

You have created secrets in the selected classification for your project.

What to do next

- You can view, edit, or delete the secrets added to a classification any time you want.
- You can use the secrets in the tests that require these secrets during test runs.

Granting access to members or member roles

You can grant Read access to the secrets added to the classification in your project at the role level for members of a designated role. Additionally, you can provide Read and Write access to specific members. Without access to the classification, members cannot view, create, edit, delete, or use the secrets added to the classification.

Before you begin

You must be a project **Owner** and have completed the following tasks:

- Created a project in Test Hub. See [Managing DevOps Test Hub projects on page 1010](#).
- Added one or more users to your project. See [Adding users to a project on page 1017](#).
- Created at least one classification. See [Creating a classification on page 193](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.

The **Security** page is displayed.

4. Click a classification in the list under **Classification** to which the secrets are added and for which you want to grant the access.

5. Select any of the following methods to grant access to a member:

Choose from:

- Select the checkbox for the specific role under **Grant Read Only Access to roles** to grant read only access to the members of a specific role. For example, if you select **Tester**, then all members in the project with a tester role are granted read only access to the selected classification. You can select any role or all the roles listed.
- Perform the following steps to grant read/write access to specific members:
 - a. Click **Add Members** inline with **Members with Read/Write Access**.

The **Additional Members** dialog is displayed.

- b. Enter the name or email ID to search the user or navigate through the list and select the member you want to grant access. You can select multiple members from the list.



Important:




Irrespective of the role that the member (**Owner**, **Tester** or **Viewer**) was assigned in the project, the access to the classification to which the secret is added must be specifically granted to the members from the **Security** page.

c. Click **Add Members**.

You granted the read/write access to the added members.

6. Select any of the following methods to revoke access to a member:

Choose from:

- Clear the checkbox for the specific role under **Grant Read Only Access to roles**, to revoke read only access to the members of a specific role. For example, if you clear the **Tester** checkbox then access for all members in the project with a tester role is revoked for the selected classification. You can select any role or all the roles listed.
- Click the **Delete** icon  inline with the member name to remove the specific members from the **Members with Read/Write Access** list.



Notes:

- Any member with access to the classification can remove access to other members specifically added or of all members with a specific role.
- Members with access to the classification can remove themselves from the access list provided that there is at least one member in the list. After removing themselves, members cannot add themselves back to the access list and must be added by any of the other members in the list.

Results

You have added members from your project or members with specific role to the access list of people who can access secrets in the selected classification, or you have removed specific members or members with specific role from the access list.

What to do next

You can add secrets to the classifications for your project.

Managing secrets

You can view, edit, or delete the secrets configured under a classification any time after you created secrets or after you were granted access to the secrets classification. You can change the value of the secret by editing the secret. You can delete secrets that you no longer require in your test environment.

Before you begin

You must have completed the following tasks:

- You must have created a project on Test Hub and configured a secrets in your project.
- You must have created secrets in the selected classification or the classifications must contain secrets.
- You must be a member with access to the secrets classification.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Security**.




The **Security** page is displayed.


4. Perform the following steps:

- Select the **Classification** from the list in the **Security** page.
- Optionally, select the classification that you want from the list if there are multiple classifications in the project.

The secrets configured in the selected classifications are displayed.

5. Complete the steps for the task you want to perform as listed in the following table:

Task	Action
Viewing a secret value	<p>Click the View icon  inline with the secret to view the entered value. The value most likely is a password for the secret.</p> <p>The value configured for the secret is displayed.</p>
Editing a secret value	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Click the Edit icon  inline with secret that you want to edit, <p>The Edit secret dialog is displayed.</p> <ol style="list-style-type: none"> Enter a new value for the secret as its <i>Value</i>. The value can be a password for the secret. <p> Note: You can only change the value of the secret.</p> <ol style="list-style-type: none"> Click Save. <p>The value of the selected secret is changed.</p>
Deleting a secret	<p>Perform the following steps:</p>

Task	Action
	<p>a. Click the Delete icon  inline with the secret that you want to delete.</p> <p>The Delete secret dialog is displayed.</p> <p>b. Click Delete.</p> <p>After you delete the secrets, the secrets listed in the classification is removed from the list.</p>

Results

- You viewed the password configured of the secret under a classification that you created or were granted access.
- You changed the secret value of the secret under a classification in your project.
- You deleted and removed the secret from the selected classification in your project.

What to do next

You can use secrets in the tests that require these secrets during test runs.

Related information

[Becoming a project member on page 1020](#)

[Test run configurations on page 446](#)

Managing notifications

HCL DevOps Test Hub (Test Hub) provides a feature to display notifications for different events within the user interface (UI) of Test Hub. You can configure an SMTP server on Test Hub when you want Test Hub to send out notifications about the different events as emails to the subscribed users. You can also configure third-party applications as webhooks on Test Hub when you want Test Hub to send push notifications about the server events to the third-party applications in a preset template.

Test Hub provides you the following features to receive notifications about server events:

Channels	Description
In-app	Notifications are displayed within the UI of Test Hub.
SMTP	Notifications are sent in emails to subscribed users of Test Hub by an SMTP server.

Channels	Description
Webhooks	Notifications in a preset template are sent as push notifications by Test Hub to the configured webhook of a third-party application, such as MS Teams, Slack, or others.

Managing notifications in the server UI

Viewing the in-app notifications or notifications in the server UI is a feature that is enabled automatically when you register as a user in Test Hub.

When you register as a user in Test Hub, the in-app notifications are automatically subscribed for you.

If you want to receive notifications about specific events and not all events, you can change the default subscriptions and modify your subscription for the server events. You can also view the in-app notifications that are displayed for you. See [Managing in-app notifications on page 1041](#).

Managing SMTP notifications

You, as a *Server Administrator* or *Team Space Owner* can configure an existing Simple Mail Transfer Protocol (SMTP) server so that Test Hub can send out notifications about the server events in emails to the registered users who have subscribed to the SMTP notifications. See [Configuring an SMTP server to manage email notifications on page 1050](#).

If you, as a registered user of Test Hub want to receive the email notifications about specific events and not all events, you can change the subscriptions for the server events. See [Managing email notifications on page 1047](#).

Managing webhook notifications

You, as a *Server Administrator*, *Team Space Owner*, or *Project Owner* can configure third-party applications as webhooks so that Test Hub can send out push notifications about the server events to the third-party applications. See [Configuration of a webhook on page 1051](#).

Managing in-app notifications

You can manage the in-app notifications by changing the subscriptions that are automatically enabled for you for all events that occur on . You can view the server events that are automatically subscribed and then change the subscription for any or all events. For example, as a team space owner, you can choose to be notified when members of the team space create new projects or delete their projects but not be notified when tests are run.

Before you begin

-
-

About this task

The in-app notifications that are automatically subscribed for you, are displayed in the server UI under the following categories for the server events that occur:

Category	Notifications displayed
User	Include information about actions performed by team space members in the team space.
System	Include information about actions performed by registered users at the server level.

- 1.
2. Select from the following actions that you want to perform:
 - Modify notification settings. Go to [Step 3 on page 1042](#).
 - View notifications. Go to [Step 4 on page 1043](#).
 - Unsubscribe from the in-app notifications. Go to [Step 5 on page 1044](#).
3. Perform the following steps to modify the notification settings:


a. Click the **User** icon  on the menu bar.

b. Click **Notification Settings**.


The **Notification Settings** dialog is displayed.

c. Change the language in which you want to receive the notifications, if you do not want to use the default language by selecting a language from the list.

The default language is the language set on the computer that hosts .

d. Identify the server events for which you want to change the subscription, and then click the Expand icon  in the row of the server event.

You can notice that for each of the server events, the in-app channel is displayed as selected, which is the default selection unless you have modified the subscription.

Notifications	Channels
Execution Completed with Verdict Error	In App 
Execution Completed with Verdict Fail	<input type="checkbox"/>
Execution Completed with Verdict Pass	<input checked="" type="checkbox"/> In App
Execution Completed without any Verdict	<input type="checkbox"/> SMTP

- e. Perform the actions indicated in the following table to clear the subscriptions for server events that you do not want to be notified and to select the in-app subscription for the server events that you want to be notified.

When	Action
You do not want to receive the in-app notifications for a specific server event.	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Expand the channel list in the row of the server event. Clear the checkbox in the In App row to clear the subscription. Click Apply. <p>You have unsubscribed from the in-app channel and the in-app notifications for the server event is not displayed for you when the event occurs.</p>
You want to receive the in-app notifications for a specific server event.	<p>Perform the following steps:</p> <ol style="list-style-type: none"> Expand the channel list in the row of the server event. Select the checkbox in the In App row, if it is cleared. Click Apply. <p>You have subscribed to the in-app channel and the in-app notifications for the server event is displayed for you when the event occurs.</p>

- f. Go to [Results on page 1046](#).

4. Perform the following steps to view the in-app notifications:

- a. Click the **Notifications** icon  on the menu bar.

The **Notifications** dialog is displayed.

The notifications are displayed in the following tabs:

- User
- System


You can find the information about the default notifications that you can view based on your role in the server as in-app notifications in the **USER** tab and **SYSTEM** tab.

Category	Notification details	or	or		
User notifications	Information about events at the server level or across team	✓	✓	✗	✗

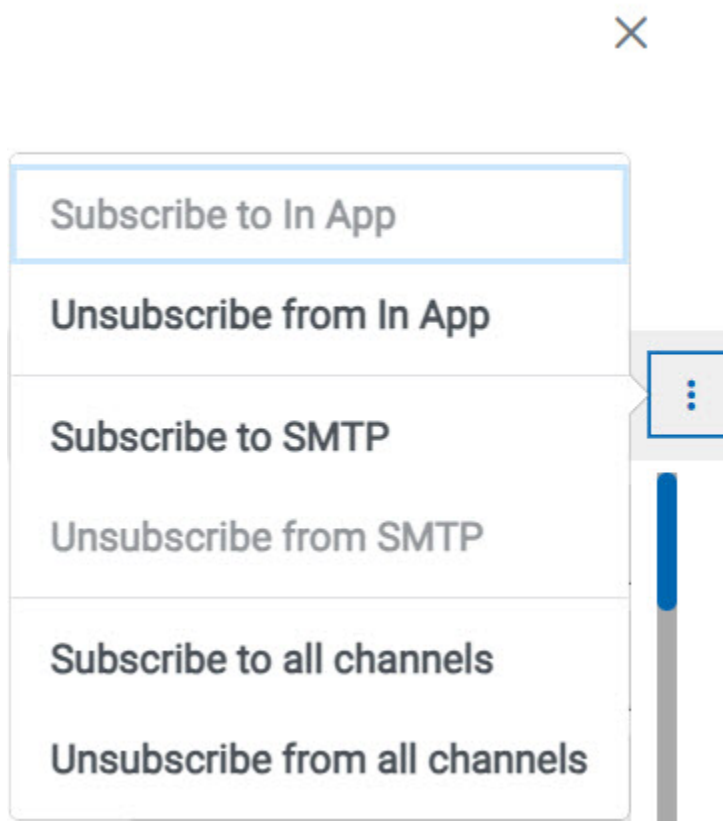
Category	Notification details	or	or		
	spaces performed by other team space owners.				
	Information about events in the team space performed by other team space members.	✓	✗	✗	✗
	Information about events in the project performed by other project members.	✗	✓	✓	✗
	Information about events performed by you in a project.	✗	✗	✓	✓
System notifications	Information about events at the server level or across team spaces.	✓	✓	✗	✗
	Information about events in the team space performed by other team space members.	✓	✓	✗	✗
	Information about events in the projects at the team space level performed by other project members.	✓	✓	✗	✗

The notifications in the **USER** tab is displayed as the default option.

- b. View the notifications in the **USER** tab.
 - c. View the notifications at the server level by clicking the **SYSTEM** tab.
 - d.
5. Perform the following steps to unsubscribe to the in-app notifications:
- a.
 - b.

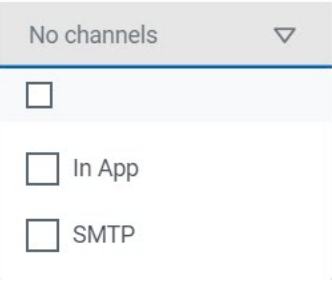
- c. Click the **Open action menu** icon  in the row of **Notifications**.

The following panel is displayed:



- d. Perform the actions indicated in the following table for the option you want:

When	Action	Results in
You do not want to receive any in-app notifications.	Click the Unsubscribe from In App option, and then click Apply .	<p>The in-app channel selection for all server events is cleared and you are unsubscribed from the in-app channel. The subscription to the SMTP channel is not changed.</p> <p>If you had subscribed to the SMTP notifications for all events, the SMTP channel is displayed as selected. If you had not subscribed to the SMTP</p>

When	Action	Results in
		channel, then the SMTP check-box is not selected.
<p>You do not want to receive any in-app or the SMTP email notifications.</p>	<p>Click the Unsubscribe from all Channels option, and then click Apply.</p>	<p>The in-app channel and the SMTP channel selections for all server events are cleared and you are unsubscribed from both the channels. No channel is displayed as selected for subscription for all server events.</p> 
<p>You want to subscribe to the in-app notifications for all server events.</p>	<p>Click the Subscribe to In App option, and then click Apply.</p>	<p>The in-app channel for all server events is selected and you are subscribed to the in-app channel. The in-app notifications is displayed as selected for subscription for all server events.</p>
<p>You want to receive notifications as in-app and as SMTP email notifications.</p>	<p>Click the Subscribe to all Channels option, and then click Apply.</p>	<p>The in-app channel and the SMTP channel are displayed as selected for all server events and you are subscribed to both the channels. However, the SMTP notifications are sent to your email only if the SMTP server is configured on .</p>

e.

Results

You completed the following tasks:

- Modified the notification subscriptions.
- Viewed the notifications displayed for you in the **USER** tab and **SYSTEM** tab.
- Unsubscribed from or subscribed to either the in-app channel or both the in-app and SMTP channels.

Related information

[Managing notifications on page 1040](#)

Managing email notifications

You can manage the email notifications by changing your subscription to the SMTP channel. For example, as a project owner, you can choose to be notified when members of the project add a repository or run a test in a team space but not be notified when projects are created in the team space.

Before you begin

You must have completed the following tasks:

- Ensured that the *Server Administrator* or *Team Space Owner* has configured a Simple Mail Transfer Protocol (SMTP) server. See [Configuring an SMTP server to manage email notifications on page 1050](#).
- Ensured that you are assigned a role as a *Member* or *Project Creator* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that you are assigned a role as a *Viewer*, *Tester*, or *Project Owner* in the project.

About this task

After the SMTP server is configured on Test Hub, you can subscribe to the SMTP channel so that you can receive email notifications for server events that occur on HCL DevOps Test Hub (Test Hub). You can view the server events that are subscribed in the **Notifications Settings** dialog. You can either unsubscribe from the SMTP channel or change your subscriptions for the server events for which you do not want the email notifications.

1. Log in to Test Hub.


The team space that contains your project is displayed.

2. Click the **User** icon  on the menu bar.
3. Click **Notification Settings**.

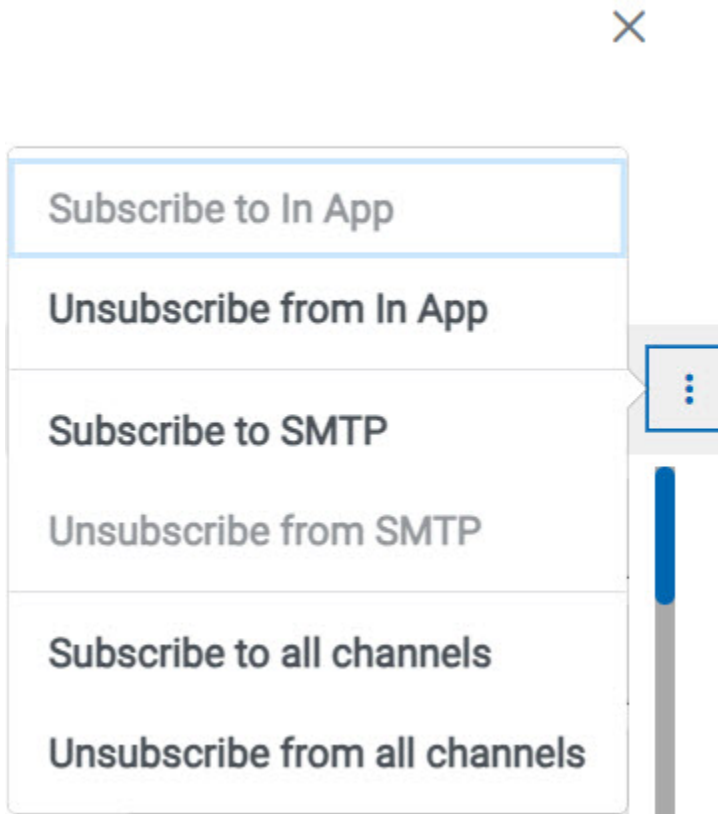
The **Notification Settings** dialog is displayed.

4. Change the language in which you want to receive the email notifications, if you do not want to use the default language by selecting a language from the list.

The default language is the language set on the computer that hosts Test Hub.

5. Click the **Open action menu** icon  in the row of **Notifications**.

The following panel is displayed:



You can notice that the **Subscribe to In App** channel is displayed as selected, unless you modified the default in-app subscription previously.

6. Perform the actions indicated in the following table for the option you want for the email notifications:

When	Action	Results in
You want to subscribe to the email notifications in the SMTP channel for all server events.	Click the Subscribe to SMTP option, and then click Apply .	The SMTP channel for all server events is selected and you are subscribed to the SMTP channel. The SMTP option is displayed as selected for subscription for all server events.
You do not want to receive any email notifications.	Click the Unsubscribe from SMTP option, and then click Apply .	The SMTP channel selection for all server events is cleared and you are unsubscribed from the SMTP channel.

When	Action	Results in
		The SMTP option is displayed as cleared for subscription for all server events.
You want to receive the email notifications for a specific server event.	Perform the following steps: a. Expand the channel list in the row of the server event. b. Select the SMTP checkbox. c. Click Apply .	You have subscribed to the SMTP channel to receive email notifications for the server event.
You do not want to receive the email notifications for a specific server event, if you subscribed to it previously.	Perform the following steps: a. Expand the channel list in the row of the server event. b. Clear the checkbox for the SMTP channel option. c. Click Apply .	You have unsubscribed from the SMTP channel for the server event and the email notifications are not sent to you when the event occurs.
You want to receive notifications as in-app and as SMTP email notifications.	Click the Subscribe to all Channels option, and then click Apply .	The in-app channel and the SMTP channel are displayed as selected for all server events and you are subscribed to both the channels.
You do not want to receive any in-app or the SMTP email notifications.	Click the Unsubscribe from all Channels option, and then click Apply .	The in-app channel and the SMTP channel selections for all server events are cleared and you are unsubscribed from both the channels. No channel is displayed as selected for subscription for all server events.

Results

You completed the following tasks:

- Subscribed to the SMTP channel for receiving email notifications for the selected server events.
- Unsubscribed from the SMTP channel for selected server events that you subscribed previously.
- Subscribed to the SMTP channel for receiving email notifications for all server events.

- Unsubscribed from the SMTP channel for receiving email notifications for all server events.
- Subscribed to or unsubscribed from the in-app and SMTP channels.

Related information

[Managing notifications on page 1040](#)

Configuring an SMTP server to manage email notifications

Before you can receive email notifications about events that occur in HCL DevOps Test Hub (Test Hub), you need to configure an existing Simple Mail Transfer Protocol (SMTP) server in Test Hub. When events occur in Test Hub, email notifications are sent to the registered users of Test Hub if they are subscribed to the SMTP notification channel.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Ensured that an existing SMTP server is available to be connected to Test Hub.

About this task

The configuration works for emails that are supported by the SMTP server. To use this capability, you must enable Test Hub to connect to the SMTP server to send email notifications.

As a server administrator, you can configure email notifications so that other users can receive information about actions on the server projects.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click the **Settings** icon  from the menu bar.

The **Notifications** page is displayed.

3. Click **Add Configuration**.
4. Provide the following details to configure the SMTP server:
 - A name to the configuration.
 - The hostname and port number of the SMTP server.



Note: The default configuration uses a local SMTP server that listens to the default port 25. This SMTP server does not require any authentication.

- The email address of the sender. The email address is displayed in email notifications. The email notifications are sent to the users based on the notification settings.
- The encryption type for secure communications is when the SMTP server is on an external network. You can enable either **STARTTLS** or **SSL/TLS**.



Note: The SMTP server for the secure communication listens to the default port 465.

- Login credentials are required if the SMTP server requires authentication.

5. Click **Add**.



Note: The SMTP configuration is added after a valid connection is established with the SMTP server.

Results

You configured an SMTP server to connect with Test Hub and send out email notifications about the server events to users who have subscribed to the SMTP channel.

What to do next

You can modify the default subscriptions for the SMTP channel, see [Managing email notifications on page 1047](#).

Related information

[Managing notifications on page 1040](#)

Configuration of a webhook

You can configure webhooks to send notifications to a messaging application such as Microsoft Teams, Slack, or any generic application that can receive an HTTP POST request when any of the events occur on HCL DevOps Test Hub (Test Hub). For example, when you add or remove a member from a team space or when you run a test.

A webhook is an HTTP custom callback that provides an application to send notifications to other applications. Webhooks help applications automatically communicate with other external applications.

The following table lists the tasks that you can perform and manage for the webhooks:

Task	Go to...
Configure a webhook template	Configuring a webhook template on page 1052

Task	Go to...
Configure a webhook	Configuring a project webhook on page 1063
Edit, clone, or delete a webhook template	Managing the webhook template configuration on page 1065
Edit, clone, or delete a webhook	Managing the webhook configuration on page 1068

Webhooks overview

Webhooks send a notification to the configured messaging applications when specific events occur on HCL DevOps Test Hub (Test Hub).

When the event occurs, Test Hub automatically sends an HTTP POST request with event-related information in a JSON format or in plain text to the URL that you configured. You can define a template body or use the sample template that is set on Test Hub. You must create a channel type, a template, and a webhook to send an HTTP POST request.

You can manage the webhook template configuration and webhook configuration on Test Hub. You can identify the events and view the status of the events that are posted on the messaging application whenever an event occurs.

Configuring a webhook template

You can configure a project, team space and a server webhook template on HCL DevOps Test Hub (Test Hub) to send notifications to the configured messaging application.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned the role of a *Project Owner* in the project. See [Managing access to the server projects on page 1021](#).
- Ensured that you are assigned a role as a *Team Space Owner* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Ensured that you are familiar with Apache Velocity. See [Apache Velocity Documentation](#).

About this task

You can create a template based on the channel type that you configured. You can select a generic template to send notifications to the generic application. You can also select a JSON sample template with specific fields defined to display event-related notifications in Microsoft Teams or Slack. You can use the sample template that is available for the messaging application or you can also define a JSON template to post the message about the events that occur on Test Hub.

1. Select from the following actions you want to perform:
 - Configure a project webhook template. Go to [Step 2 on page 1053](#).
 - Configure a team space webhook template. Go to [Step 3 on page 1055](#).
 - Configure a server webhook template. Go to [Step 4 on page 1057](#).
2. Perform the following steps to configure a project webhook template.

- a. Log in to Test Hub.

The team space that contains your project is displayed.

- b. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.


- c. Click **Manage > Webhooks**.

The **TEMPLATES** tab of Webhooks is displayed.

- d. Click **New Template**.

The **New template** page is displayed.

- e. Perform the following actions on the **DETAILS** tab:

- i. Enter a unique name for the template.
- ii. Click the **Channel type** option, and then select an existing channel type.
- iii. Alternatively, you can create a new channel type by clicking **Create new channel type** icon  if there are no webhook templates available in the drop-down list.

The **New channel type** dialog is displayed.

- iv. Enter a unique name for the channel type, and then enter a description for the channel type.



Note: You can also create a new channel type or edit a channel type from **Webhooks > TEMPLATES > Edit channel type** option.

- f. Perform the following steps if you want to select either a channel icon or a brand icon:

- i. Click the **Channel type** icon .

The **Select icon** page is displayed.

- ii. View a list of icons that are displayed in the **GENERAL** tab.
- iii. Search for the icons by entering any text contained in the icon name in the **Search** field, and then select the required icon.

The selected icon is displayed in the **Selected icon** box.

- iv. Click the **BRANDS** tab.

View a list of icons that are displayed.

- v. Search for the brands by entering any text contained in the brand name in the **Search** field, and then select the brand.

The selected brand is displayed in the **Selected icon** box.

- vi. Click **Ok**.

The default icon is replaced with the selected icon.

- g. Enter a description for the channel type, and then click **Apply**.

The channel type is created and added to the **Channel Type** list.

- h. Enter a description for the template.

- i. Perform the following steps to select an event:

- i. Click **This template is suitable for events** and select All Events, or any of the subset events from the following list:

- All Events
- Team Space Events
- Project Events
- Execution Events
- License Events


- ii. Search for the events by entering any text contained in the events name in the **Search** field, and then select the corresponding checkbox to select the specific event.



Note: All Events is the default event that is set for the webhook template.

- j. Click the **TEMPLATES** tab, and then perform any of the following actions:

- i. Perform the following steps to insert a generic template:


1. Click the **cloud** icon  to insert a velocity text processing template in the **Template body**.

A message is displayed about the action that inserts a text in the template body.

2. Click **Ok**.

The text is inserted in the **Template body**.

- ii. Perform the following steps to insert a Microsoft Teams sample template:


1. Click the **Microsoft Teams** icon  to replace a template with a Microsoft Teams template.

A message is displayed about the action that replaces the content of the template body with a Microsoft Teams template.

2. Click **Ok**.

A Microsoft Teams sample template is inserted in the **Template body**.

iii. Perform the following steps to insert a Slack sample template:

1. Click the **Slack** icon  .
2. A message is displayed about the action that replaces the content of the template body with a Slack template.
3. Click **Ok**.

A Slack sample template is inserted in the **Template body**.

k. Click **Apply**.

The template is created and added to the **TEMPLATES** tab.

3. Perform the following steps to configure a team space webhook template:

a. Log in to Test Hub.

The team space that contains your project is displayed.


b. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

c. Click **New Template**.

The **New template** page is displayed.

d. Perform the following actions on the **DETAILS** tab:

- i. Enter a unique name for the template.
- ii. Click the **Channel type** option, and then select an existing channel type.
- iii. Alternatively, you can create a new channel type by clicking **Create new channel type** icon  if there are no webhook templates available in the drop-down list.

The **New channel type** dialog is displayed.

iv. Enter a unique name for the channel type, and then enter a description for the channel type.



Note: You can also create a new channel type or edit a channel type from **Webhooks > TEMPLATES > Edit channel type** option.

e. Perform the following steps if you want to select either a channel icon or a brand icon:

- i. Click the **Channel type** icon  .

The **Select icon** page is displayed.

- ii. View a list of icons that are displayed in the **GENERAL** tab.
- iii. Search for the icons by entering any text contained in the icon name in the **Search** field, and then select the required icon.

The selected icon is displayed in the **Selected icon** box.

- iv. Click the **BRANDS** tab.

View a list of icons that are displayed.

- v. Search for the brands by entering any text contained in the brand name in the **Search** field, and then select the brand.

The selected brand is displayed in the **Selected icon** box.

- vi. Click **Ok**.

The default icon is replaced with the selected icon.

- f. Enter a description for the channel type, and then click **Apply**.

The channel type is created and added to the **Channel Type** list.

- g. Enter a description for the template.

- h. Perform the following steps to select an event:

- i. Click **This template is suitable for events** and select All Events, or any of the subset events from the following list:

- All Events
- Team Space Events
- Project Events
- Execution Events
- License Events


- ii. Search for the events by entering any text contained in the events name in the **Search** field, and then select the corresponding checkbox to select the specific event.



Note: All Events is the default event that is set for the webhook template.

- i. Click the **TEMPLATES** tab, and then perform any of the following actions:

- i. Perform the following steps to insert a generic template:


1. Click the **cloud** icon  to insert a velocity text processing template in the **Template body**.

A message is displayed about the action that inserts a text in the template body.

2. Click **Ok**.

The text is inserted in the **Template body**.

- ii. Perform the following steps to insert a Microsoft Teams sample template:


1. Click the **Microsoft Teams** icon  to replace a template with a Microsoft Teams template.

A message is displayed about the action that replaces the content of the template body with a Microsoft Teams template.

2. Click **Ok**.

A Microsoft Teams sample template is inserted in the **Template body**.

- iii. Perform the following steps to insert a Slack sample template:

1. Click the **Slack** icon  .
2. A message is displayed about the action that replaces the content of the template body with a Slack template.
3. Click **Ok**.

A Slack sample template is inserted in the **Template body**.

- j. Click **Apply**.

The template is created and added to the **TEMPLATES** tab.

4. Perform the following steps to configure a server webhook template:

- a. Log in to Test Hub.

The team space that contains your project is displayed.

- b. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

- c. Click the **Settings**  icon on the menu bar.


- d. Click **Webhooks**.

The **TEMPLATES** tab of Webhooks is displayed.

- e. Click **New Template**.

The **New template** page is displayed.

- f. Perform the following actions on the **DETAILS** tab:

- i. Enter a unique name for the template.
- ii. Click the **Channel type** option, and then select an existing channel type.
- iii. Alternatively, you can create a new channel type by clicking **Create new channel type** icon  if there are no webhook templates available in the drop-down list.

The **New channel type** dialog is displayed.

- iv. Enter a unique name for the channel type, and then enter a description for the channel type.



Note: You can also create a new channel type or edit a channel type from **Webhooks > TEMPLATES > Edit channel type** option.

- g. Perform the following steps if you want to select either a channel icon or a brand icon:

- i. Click the **Channel type** icon .

The **Select icon** page is displayed.

- ii. View a list of icons that are displayed in the **GENERAL** tab.
- iii. Search for the icons by entering any text contained in the icon name in the **Search** field, and then select the required icon.

The selected icon is displayed in the **Selected icon** box.

- iv. Click the **BRANDS** tab.

View a list of icons that are displayed.

- v. Search for the brands by entering any text contained in the brand name in the **Search** field, and then select the brand.

The selected brand is displayed in the **Selected icon** box.

- vi. Click **Ok**.

The default icon is replaced with the selected icon.

- h. Enter a description for the channel type, and then click **Apply**.

The channel type is created and added to the **Channel Type** list.

- i. Perform the following steps to select an event:

- i. Click **This template is suitable for events** and select All Events, or any of the subset events from the following list:

- All Events
- Team Space Events
- Project Events
- Execution Events
- License Events


- ii. Search for the events by entering any text contained in the events name in the **Search** field, and then select the corresponding checkbox to select the specific event.



Note: All Events is the default event that is set for the webhook template.

- j. Click the **TEMPLATES** tab, and then perform any of the following actions:

i. Perform the following steps to insert a generic template:


1. Click the **cloud** icon  to insert a velocity text processing template in the **Template body**.

A message is displayed about the action that inserts a text in the template body.

2. Click **Ok**.

The text is inserted in the **Template body**.

ii. Perform the following steps to insert a Microsoft Teams sample template:


1. Click the **Microsoft Teams** icon  to replace a template with a Microsoft Teams template.

A message is displayed about the action that replaces the content of the template body with a Microsoft Teams template.

2. Click **Ok**.

A Microsoft Teams sample template is inserted in the **Template body**.

iii. Perform the following steps to insert a Slack sample template:

1. Click the **Slack** icon  .
2. A message is displayed about the action that replaces the content of the template body with a Slack template.
3. Click **Ok**.

A Slack sample template is inserted in the **Template body**.

k. Click **Apply**.

The template is created and added to the **TEMPLATES** tab.

Results

You have configured and created a project, team space and a server webhook template on Test Hub.

What to do next

You must configure a webhook to push the events from Test Hub to the messaging application. See [Configuring a project webhook on page 1063](#).

Configuring a server webhook

You can configure a server webhook to send notifications to a messaging application whenever an event occurs on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role of a *Server Administrator*. See [Managing access to the server projects on page 1021](#).
- Ensured that you have access to the messaging applications.
- Created a channel type.
- Configured your messaging application.
- Copied the URL of the external communication application. See [Creating an incoming webhook](#).
- Created a webhook template. See [Configuring a webhook template on page 1052](#).

About this task

When you configure a webhook on Test Hub, you must copy the URL of the communication application in the **Webhook URL** field and paste the outgoing URL, and then select the events. You can select either all events or a subset of events.

You can be notified of the following types of events:


- All Events
- Team space Events
- Project Events
- Execution Events
- License Events

You can configure the webhook resume policy by selecting either the Queue old messages or Discard old messages option. When a server webhook is disabled due to communication failures or network issues, the server event notifications are not delivered to the configured webhook, and you receive a notification that the webhook is disabled. By configuring the queue old messages, you can ensure that the event notifications are not lost and are delivered to the configured webhook after the server webhook is enabled again.

If you configure the Discard old messages for a server webhook, the event notifications are discarded from the moment the webhook is disabled. Only server events that occur after the webhook is re-enabled are sent to the configured webhook.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click the **Settings** icon  on the menu bar, and then click **Webhooks**.


The **TEMPLATES** tab of Webhooks is displayed.

3. Click the **WEBHOOKS** tab, and then click **New Webhook**.

The **New Webhook** page is displayed.



4. Enter a name for the webhook and paste the URL that you copied in the **Webhook URL** field.

You must copy the URL while you configure your communication application.

5. Click the **Webhook resume policy**, and then select a resume policy option from the drop-down list. You can select any of the following options:
 - a. Queue old messages
 - b. Discard old messages
6. Click the **Channel Type**, and then select a channel type from the list.
7. Enter a description for the webhook, and then select a template from the drop-down list.
A corresponding event that you configured for the template is displayed in the **Events** field.
8. Click the **Templates** icon  [Template](#), to add the templates that are configured for the channel type.



Note: If you want to change the events, then click the events option and select the required events.

9. You can perform any of the following steps:
 - a. Click move the template to a higher priority icon  to move the template to a higher priority.
 - b. Click move the template to a lower priority icon  to move the template to a lower priority.
10. Click **Apply**.
The webhook is created and added to the **WEBHOOKS** tab.

Results

You have configured and created a server webhook on Test Hub.

What to do next

You can perform any of the following tasks:

- You can duplicate or delete the template that you created on Test Hub. See [Managing the webhook template configuration on page 1065](#).
- You can duplicate or delete the webhook you created on Test Hub. See [Managing the webhook configuration on page 1068](#).

Configuring a team space webhook

You can configure a team space webhook to send notifications to a messaging application whenever an event occurs on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned a role as a *Team Space Owner* in the team space. See [Managing members and their roles in a team space on page 1006](#).

- Ensured that you have access to the messaging applications.
- Created a channel type.
- Configured your messaging application.
- Copied the URL of the external communication application. See [Creating an incoming webhook](#).
- Created a webhook template. See [Configuring a webhook template on page 1052](#).

About this task

When you configure a webhook on Test Hub, you must copy the URL of the communication application in the **Webhook URL** field and paste the outgoing URL, and then select the events. You can select either all events or a subset of events.

You can be notified of the following types of events:


- All Events
- Team space Events
- Project Events
- Execution Events
- License Events

You can configure the webhook resume policy by selecting either the Queue old messages or Discard old messages option. When a team space webhook is disabled due to communication failures or network issues, the team space event notifications are not delivered to the configured webhook, and you receive a notification that the webhook is disabled. By configuring the queue old messages, you can ensure that the event notifications are not lost and are delivered to the configured webhook after the team space webhook is enabled again.

If you configure the Discard old messages for a team space webhook, the event notifications are discarded from the moment the webhook is disabled. Only team space events that occur after the webhook is re-enabled are sent to the configured webhook.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click the **Settings** icon  on the left navigation pane, and then click **Manage Team space**.

3. Click **Webhooks**.

The **TEMPLATES** tab of Webhooks is displayed.


4. Click the **WEBHOOKS** tab, and then click **New Webhook**.

The **New Webhook** page is displayed.

5. Enter a name for the webhook and paste the URL that you copied in the **Webhook URL** field.



You must copy the URL while you configure your communication application.

6. Click the **Webhook resume policy**, and then select a resume policy option from the drop-down list. You can select any of the following options:

- a. Queue old messages
 - b. Discard old messages
7. Click the **Channel Type**, and then select a channel type from the list.
 8. Enter a description for the webhook, and then select a template from the drop-down list.
A corresponding event that you configured for the template is displayed in the **Events** field.
 9. Click the **Templates** icon  [Template](#), to add the templates that are configured for the channel type.



Note: If you want to change the events, then click the events option and select the required events.

10. You can perform any of the following steps:
 - a. Click move the template to a higher priority icon  to move the template to a higher priority.
 - b. Click move the template to a lower priority icon  to move the template to a lower priority.
11. Click **Apply**.
The webhook is created and added to the **WEBHOOKS** tab.

Results

You have configured and created a team space webhook on Test Hub.

What to do next

- You can duplicate or delete the template that you created on Test Hub. See [Managing the webhook template configuration on page 1065](#).
- You can duplicate or delete the webhook you created on Test Hub. See [Managing the webhook configuration on page 1068](#).

Configuring a project webhook

You can configure a project webhook to send notifications to a messaging application whenever an event occurs on HCL DevOps Test Hub (Test Hub).

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned the role of a *Project Owner* in the project. See [Managing access to the server projects on page 1021](#).
- Created a project in your team space in Test Hub. See [Adding a project on page 1013](#).
- Ensured that you have access to the messaging applications.
- Created a channel type.
- Configured your messaging application.
- Copied the URL of the external communication application. See [Creating an incoming webhook](#).
- Created a webhook template. See [Configuring a webhook template on page 1052](#).

About this task

When you configure a webhook on Test Hub, you must copy the URL of the communication application in the **Webhook URL** field and paste the outgoing URL, and then select the events. You can select either all events or a subset of events.

You can be notified of the following types of events:

- All Events
- Team space Events
- Project Events
- Execution Events
- License Events

You can configure the webhook resume policy by selecting either the Queue old messages or Discard old messages option. When a project webhook is disabled due to communication failures or network issues, the project event notifications are not delivered to the configured webhook, and you receive a notification that the webhook is disabled. By configuring the queue old messages, you can ensure that the event notifications are not lost and are delivered to the configured webhook after the project webhook is enabled again.

If you configure the Discard old messages for a project webhook, the event notifications are discarded from the moment the webhook is disabled. Only project events that occur after the webhook is re-enabled are sent to the configured webhook.

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

3. Click **Manage > Webhooks** in the navigation pane.

The **Webhooks** page is displayed.

4. Click the **WEBHOOKS** tab, and then click **New Webhook**.

The **New Webhook** page is displayed.

5. Enter a name for the webhook and paste the URL that you copied in the **Webhook URL** field.


You must copy the URL while you configure your communication application.

6. Click the **Webhook resume policy**, and then select a resume policy option from the drop-down list. You can select any of the following options:

- a. Queue old messages
- b. Discard old messages



The Queue old messages is selected as the default option.

7. Click the **Channel Type**, and then select a channel type from the list.

8. Enter a description for the webhook, and then select a template from the drop-down list.
A corresponding event that you configured for the template is displayed in the **Events** field.
9. Click the **Templates**  [Template](#) icon, to add the templates that are configured for the channel type.



Note: If you want to change the events, then click the events option and select the required events.

10. You can perform any of the following steps:
 - a. Click move the template to a higher priority icon  to move the template to a higher priority.
 - b. Click move the template to a lower priority icon  to move the template to a lower priority.
11. Click **Apply**.
The webhook is created and added to the **WEBHOOKS** tab.

Results

You have configured and created a project webhook on Test Hub.

What to do next

You can perform any of the following tasks:

- You can duplicate or delete the template that you created on Test Hub. See [Managing the webhook template configuration on page 1065](#).
- You can duplicate or delete the webhook you created on Test Hub. See [Managing the webhook configuration on page 1068](#).

Managing the webhook template configuration

You can manage the webhook template configuration that you set on HCL DevOps Test Hub (Test Hub) when you want to edit, duplicate, delete the existing webhook templates, or create new templates. Before you create a webhook, you can view the templates created at server, team space, and project levels. Additionally when you create a project, team space or server webhook you can utilize the corresponding templates associated with them.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned the role of a *Project Owner* in the project. See [Managing access to the server projects on page 1021](#).
- Ensured that you are assigned a role as a *Team Space Owner* in the team space. See [Managing members and their roles in a team space on page 1006](#).
- Configured a webhook template. See [Configuring a webhook template on page 1052](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.





2. Click **Active projects > My projects > project_name** to open the project that contains the test assets.





The **Overview** page of the project is displayed.

3. Click **Manage > Webhooks** in the navigation pane.

The **TEMPLATES** tab of Webhooks is displayed.

4. Perform the tasks indicated in the following table:

Task	Action
Configuring a webhook template	Click the New Template option to configure a webhook template. See Configuring a webhook template on page 1052 .
Viewing webhook templates	<p>You can view the following details about the project and team space templates:</p> <ol style="list-style-type: none"> a. The Show all levels option enabled. b. The following icons that display the server, team space, and project webhook templates in the From column. <ul style="list-style-type: none"> ▪ Project template icon  ▪ Team space template icon  ▪ Serve template icon  <p>As a project owner, you can view project, team space, and server templates on the project template page. Similarly, as a team space owner, you can view only team space and server templates on the team space template page.</p> <p>Perform the following steps to view project or team space templates:</p> <ol style="list-style-type: none"> a. Click the Hide other levels option. <p>You can only view project webhook templates, if you are on the project template page, and you can only view team space templates, if you are on the team space template page.</p> <ol style="list-style-type: none"> b. Click the Action icons to edit, duplicate view or delete the templates. <p> Note: As a <i>Project Owner</i>, you can edit the project webhook templates; however, you can only view and clone the details of server webhook templates or team space webhook templates. If you are a <i>Team Space Owner</i> you can edit the team space templates;</p>

Task	Action
	 however, you can only view and clone the details of server web-hook templates. If you clone a webhook template, then you can edit or delete a webhook template.
Editing a webhook template	<p>Perform the following steps to edit the webhook template:</p> <ol style="list-style-type: none"> Click the Edit template icon . <p>The Edit template page is displayed.</p> <ol style="list-style-type: none"> In the Details tab, you can enter a new name for the webhook template, or select any other value for the options. In the Templates tab, you can also edit or select a template based on the channel type you selected on Details tab. Click Apply. <p>The edited template is saved to the TEMPLATES tab.</p>
Cloning a webhook template	<p>Perform the following steps to duplicate the webhook template:</p> <ol style="list-style-type: none"> Click the Duplicate template icon . <p>The Duplicate template page is displayed.</p> <ol style="list-style-type: none"> In the Details tab, you can edit or modify the values by entering a new name for the webhook template, by selecting a Channel Type, or Events. In the Templates tab, you can edit or select other webhook template based on the channel type you selected on Details tab. Click Apply. <p>The cloned webhook template is added to the TEMPLATES tab.</p>
Deleting a webhook template	<p>Perform the following steps to delete the webhook template:</p> <ol style="list-style-type: none"> Click the Delete Template icon . <p>The Delete template confirmation dialog is displayed.</p> <ol style="list-style-type: none"> Select the I understand template deletion is permanent checkbox, and then click Delete. <p>The webhook template is removed from the template list.</p>

Results

You achieved the following results:

- Edited and saved a webhook template configuration.
- Cloned a webhook template configuration.
- Deleted a webhook template configuration.

Managing the webhook configuration

You can manage the webhook configuration that you set on HCL DevOps Test Hub (Test Hub) you want to edit, duplicate, delete the existing webhook configurations, or create a webhook.

Before you begin

- Ensured that you are assigned a role of a project *Owner* in the project. See [Managing access to the server projects on page 1021](#).
- Ensured that you configured a webhook. See [Configuring a project webhook on page 1063](#).

1. Log in to Test Hub.

The team space that contains your project is displayed.

2. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.


The **Overview** page of the project is displayed.



3. Click **Manage > Webhooks** in the navigation pane.

The **Webhooks** page is displayed.

4. Click the **WEBHOOKS** tab.

5. Perform the tasks indicated in the following table:

Task	Action
New Webhook	Click the New Webhook option to configure and create a webhook. See Configuring a project webhook on page 1063 .
Editing a webhook	Perform the following steps to edit the webhook: <ol style="list-style-type: none"> a. Click the Edit Webhook icon . The Edit webhook page is displayed. b. Enter a new name for the webhook. c. Replace the webhook URL by pasting a different URL of another messaging channel you configured. d. Select another channel type to post the notification of an event. e. Select a different template, and then select events from the Events list. f. Click Close.
Cloning a webhook	Perform the following steps to clone the webhook:

Task	Action
	<p>a. Click the Duplicate webhook icon .</p> <p>The Duplicate webhook page is displayed.</p> <p>b. Enter a new name for the webhook, and then click Apply.</p> <p>The cloned webhook is added to the WEBHOOKS tab.</p>
Deleting a webhook	<p>Perform the following steps to delete the webhook:</p> <p>a. Click the Delete Webhook icon .</p> <p>The Delete Webhook confirmation dialog is displayed.</p> <p>b. Select the I understand webhook deletion is permanent checkbox, and then click Delete.</p> <p>The webhook is removed from the webhook list.</p>

Results

You achieved the following results:

- Edited and saved a webhook configuration.
- Cloned a webhook configuration.
- Deleted a webhook configuration.

Enabling a webhook

You can enable the webhook when push notifications about events occurring on HCL DevOps Test Hub (Test Hub) are not delivered to the configured webhook due to communication failures or network issues, and you receive a notification that the webhook is disabled.

Before you begin

You must have completed the following tasks:

- Ensured that you are assigned the role of a *Server Administrator*, *Team Space Owner* or a *Project Owner*. If you are a server administrator, you can have three roles-server admin, team space owner, and project owner.
- Configured a webhook. See [Configuring a project webhook on page 1063](#).

About this task

When a webhook is disabled due to communication failure or network issues, you are notified about the disablement of the webhook. Consequently notifications for the events are not delivered to the configured webhook and instead

be queued for sending after you re-enable the webhook. You have the option to either send the queued messages or discard the queued messages.

1. Select from the following actions that you want to perform:
 - Enable a project webhook. Go to [Step 2 on page 1070](#).
 - Enable a team space webhook. Go to [Step 3 on page 1070](#).
 - Enable a server webhook. Go to [Step 4 on page 1070](#)
2. Perform the following steps to manually enable a project webhook:

- a. Log in to Test Hub.

The team space that contains your project is displayed.

- b. Click **Active projects > My projects > *project_name*** to open the project that contains the test assets.

The **Overview** page of the project is displayed.

- c. Click **Manage > Webhooks**.

The **TEMPLATES** tab of Webhooks is displayed.


- d. Click the **Webhooks** tab, and then click the **Disabled** webhook option to enable a webhook that is disabled.

The event notifications that were queued when the webhook was disabled are delivered to the configured webhook or discarded by Test Hub. This setting is based on the resume policy configuration that you set for the specific project webhook.

3. Perform the following steps to manually enable a team space webhook:

- a. Log in to Test Hub.

The team space that contains your project is displayed.

- b. Click the **Settings**  icon on the navigation pane, and then click **Manage team space**.

- c. Click **Webhooks**.

The **TEMPLATES** tab of Webhooks is displayed.


- d. Click the **Webhooks** tab, and then click the **Disabled** webhook option to enable a webhook that is disabled.

The event notifications that were queued when the webhook was disabled are delivered to the configured webhook or discarded by Test Hub. This setting is based on the resume policy configuration that you set for the team space webhook.

4. Perform the following steps to manually enable a server webhook:

- a. Log in to Test Hub.

The team space that contains your project is displayed.

- b. Click the **Settings**  icon on the menu bar, and then click **Webhooks**.

The **TEMPLATES** tab of Webhooks is displayed.

- c. Click the **Webhooks** tab, and then click the **Disabled** webhook option to enable a webhook that is disabled.

The event notifications that were queued when the webhook was disabled are delivered to the configured webhook or discarded by Test Hub. This setting is based on the resume policy configuration that you set for the server webhook.

Results

You enabled the webhook on Test Hub to push notifications of the events that failed to be delivered to the configured webhook.

Management of events

When an identified event occurs on HCL DevOps Test Hub (Test Hub), a webhook sends an HTTP POST request to the URL that you configured for the webhook. The request contains the message associated with the event, in JSON format or plain text. Each event corresponds to a set of actions that occur. For example, when a member is added to a Team space or when a Test failed with an error.

The following tables describe different events that are identified on Test Hub.

Table 54. Main

Event	Description
All Events	When any of the events occur, a webhook sends a notification.

Table 55. Team Space Events

Event	Description
Team Space Management	When events occur at the team space management level, a webhook sends a notification.
Team Space Created	When a team space is created, a webhook sends a notification.
Team Space Deleted	When a team space is deleted, a webhook sends a notification
Team Space Updated	When a team space is updated, a webhook sends a notification Following are the subsets of the team space events:

Table 55. Team Space Events (continued)

Event	Description							
	<table border="1"> <thead> <tr> <th data-bbox="821 321 1045 363">Event</th> <th data-bbox="1058 321 1414 363">Notification sent</th> </tr> </thead> </table>	Event	Notification sent					
	Event	Notification sent						
	Team Space URL alias Updated	When the team space URL alias is updated.						
	Team Space Re-named	When the name of the team space is changed.						
	Team Space Be-comes Permissive	When the team space is accessible.						
	Team Space Be-comes Private	When an access to the team space is not shared.						
	Team Space Be-comes Public	When all members are provided access to the team space.						
Team Space Be-comes Restrictive	When access to the team space is limited.							
Team Space Repositories	<p>When an event occurs in the team space repositories, a webhook sends a notification.</p> <p>Following are the subsets of the team space repository events:</p> <table border="1" data-bbox="821 1180 1414 1434"> <thead> <tr> <th data-bbox="821 1190 1117 1232">Event</th> <th data-bbox="1123 1190 1414 1232">Notification sent</th> </tr> </thead> <tbody> <tr> <td data-bbox="821 1243 1117 1337">Team Space Repositories Created</td> <td data-bbox="1123 1243 1414 1337">When the team space repositories are created.</td> </tr> <tr> <td data-bbox="821 1341 1117 1436">Team Space Repositories Deleted</td> <td data-bbox="1123 1341 1414 1436">When the team space repositories are deleted.</td> </tr> </tbody> </table>		Event	Notification sent	Team Space Repositories Created	When the team space repositories are created.	Team Space Repositories Deleted	When the team space repositories are deleted.
Event	Notification sent							
Team Space Repositories Created	When the team space repositories are created.							
Team Space Repositories Deleted	When the team space repositories are deleted.							
Team Space Member	<p>When an event occurs in the team space, a webhook sends a notification.</p> <p>Following are the subsets of the team space member events:</p> <table border="1" data-bbox="821 1669 1414 1810"> <thead> <tr> <th data-bbox="821 1675 1117 1717">Event</th> <th data-bbox="1123 1675 1414 1717">Notification sent</th> </tr> </thead> <tbody> <tr> <td data-bbox="821 1728 1117 1810">Team Space Member Added</td> <td data-bbox="1123 1728 1414 1810">When a member is added to the team space.</td> </tr> </tbody> </table>		Event	Notification sent	Team Space Member Added	When a member is added to the team space.		
Event	Notification sent							
Team Space Member Added	When a member is added to the team space.							

Table 55. Team Space Events (continued)

Event	Description									
	<table border="1"> <thead> <tr> <th data-bbox="815 312 1110 357">Event</th> <th data-bbox="1110 312 1414 357">Notification sent</th> </tr> </thead> </table>	Event	Notification sent							
Event	Notification sent									
	Team Space Member Re-removed	When a member is removed from the team space.								
	Team Space Member Role Added	When a specific role for a member is added.								
	Team Space Member Role Removed	When a specific role of a member is removed.								
Team Space Role Request	<p>When a request is sent to join a team space, a webhook sends a notification.</p> <p>Following are the subsets of the team space role request events:</p> <table border="1"> <thead> <tr> <th data-bbox="815 926 1117 978">Event</th> <th data-bbox="1117 926 1414 978">Notification sent</th> </tr> </thead> <tbody> <tr> <td data-bbox="815 978 1117 1125">Team Space Role Requested</td> <td data-bbox="1117 978 1414 1125">When a member sends a request to get a role in the team space.</td> </tr> <tr> <td data-bbox="815 1125 1117 1262">Team Space Role Request Accepted</td> <td data-bbox="1117 1125 1414 1262">When the request of the member for a role in the team space is accepted.</td> </tr> <tr> <td data-bbox="815 1262 1117 1394">Team Space Role Request Rejected</td> <td data-bbox="1117 1262 1414 1394">When the request of the member for a role in the team space is rejected.</td> </tr> </tbody> </table>		Event	Notification sent	Team Space Role Requested	When a member sends a request to get a role in the team space.	Team Space Role Request Accepted	When the request of the member for a role in the team space is accepted.	Team Space Role Request Rejected	When the request of the member for a role in the team space is rejected.
Event	Notification sent									
Team Space Role Requested	When a member sends a request to get a role in the team space.									
Team Space Role Request Accepted	When the request of the member for a role in the team space is accepted.									
Team Space Role Request Rejected	When the request of the member for a role in the team space is rejected.									

Table 56. Project Events

Event	Notification sent					
Project Management	<p>When events occur at the project level, a webhook sends a notification.</p> <p>The Following are the subsets of project management events:</p> <table border="1"> <thead> <tr> <th data-bbox="815 1759 1117 1812">Event</th> <th data-bbox="1117 1759 1417 1812">Notification sent</th> </tr> </thead> <tbody> <tr> <td data-bbox="815 1812 1117 1860">Project Archived</td> <td data-bbox="1117 1812 1417 1860">When a project is archived.</td> </tr> </tbody> </table>		Event	Notification sent	Project Archived	When a project is archived.
Event	Notification sent					
Project Archived	When a project is archived.					

Table 56. Project Events (continued)

Event	Notification sent											
	<table border="1"> <thead> <tr> <th data-bbox="818 312 1110 357">Event</th> <th data-bbox="1110 312 1416 357">Notification sent</th> </tr> </thead> <tbody> <tr> <td data-bbox="818 357 1110 409">Project Created</td> <td data-bbox="1110 357 1416 409">When a project is created.</td> </tr> <tr> <td data-bbox="818 409 1110 462">Project Renamed</td> <td data-bbox="1110 409 1416 462">When a project is renamed.</td> </tr> <tr> <td data-bbox="818 462 1110 514">Project Deleted</td> <td data-bbox="1110 462 1416 514">When a project is deleted.</td> </tr> <tr> <td data-bbox="818 514 1110 619">Project Unarchived</td> <td data-bbox="1110 514 1416 619">When a project is unarchived.</td> </tr> </tbody> </table>	Event	Notification sent	Project Created	When a project is created.	Project Renamed	When a project is renamed.	Project Deleted	When a project is deleted.	Project Unarchived	When a project is unarchived.	
Event	Notification sent											
Project Created	When a project is created.											
Project Renamed	When a project is renamed.											
Project Deleted	When a project is deleted.											
Project Unarchived	When a project is unarchived.											
Project Repositories	<p>When an event occurs in the project repositories, a web-hook sends a notification.</p> <p>Following are the subsets of the project repositories events:</p> <table border="1"> <thead> <tr> <th data-bbox="818 858 1117 911">Event</th> <th data-bbox="1117 858 1416 911">Notification sent</th> </tr> </thead> <tbody> <tr> <td data-bbox="818 911 1117 1016">Project Repository Created</td> <td data-bbox="1117 911 1416 1016">When a project repository is created.</td> </tr> <tr> <td data-bbox="818 1016 1117 1110">Project Repository Deleted</td> <td data-bbox="1117 1016 1416 1110">When a project repository is deleted.</td> </tr> </tbody> </table>		Event	Notification sent	Project Repository Created	When a project repository is created.	Project Repository Deleted	When a project repository is deleted.				
Event	Notification sent											
Project Repository Created	When a project repository is created.											
Project Repository Deleted	When a project repository is deleted.											
Project Member Role	<p>When the role of a member of a project is changed, a web-hook sends a notification.</p> <p>Following are the subsets of the project member role events:</p> <table border="1"> <thead> <tr> <th data-bbox="818 1341 1117 1394">Event</th> <th data-bbox="1117 1341 1416 1394">Notification sent</th> </tr> </thead> <tbody> <tr> <td data-bbox="818 1394 1117 1499">Project Member Role Added</td> <td data-bbox="1117 1394 1416 1499">When a project member role is added to the project.</td> </tr> <tr> <td data-bbox="818 1499 1117 1635">Project Member Role Re-moved</td> <td data-bbox="1117 1499 1416 1635">When a project member role is removed from the project.</td> </tr> </tbody> </table>		Event	Notification sent	Project Member Role Added	When a project member role is added to the project.	Project Member Role Re-moved	When a project member role is removed from the project.				
Event	Notification sent											
Project Member Role Added	When a project member role is added to the project.											
Project Member Role Re-moved	When a project member role is removed from the project.											
Project Member Role Requests	<p>When a request is sent to add a member role to the project, a webhook sends a notification.</p> <p>Following are the subsets of the project member role requests events:</p>											

Table 56. Project Events (continued)

Event	Notification sent	
	Event	Notification sent
	Project Member Role Re- quested	When a member sends a request to join a project.
	Project Member Role Re- quest Accepted	When a request of the member to join a project is accepted.
	Project Member Rejected	When a request of the member to join a project is rejected.

Table 57. Execution Events

Event	Description	
Test and Suite Sta- tus	Webhook sends notifications about the status of the test and test suite. Following are the subsets of the test and test suite status events:	
	Event	Notification sent
	Execution Started	When you start to run a test or a test suite.
	Execution Stopped Manually	When the test run is manually stopped.
	Execution Completed with Verdict Pass	When the test run is completed with a pass verdict.
Test and Suite Sta- tus Errors	Webhook sends notifications about the error status of the test and test suite. Following are the subsets of test and test suite status error events:	
	Event	Notification sent
	Execution Completed with Verdict Fail	When the test run is completed with a fail ver- dict.
	Execution Completed with Verdict Error	When the test run is completed with an error status.
	Execution Completed without any Verdict	When the test run is completed without any results.
	Execution Launch Failed	When the test launch fails.
	Execution Stopped with Error	When the test run stops with an error.

Table 57. Execution Events (continued)

Event	Description									
	Event	Notification sent								
	Execution Stopped Automatically	When automatically the test run stops.								
	Execution Incomplete	When the test run is not completed.								
Virtual Service Status	<p>Webhook sends notifications about the status of the virtual services.</p> <p>Following are the subsets of the virtual service status events:</p> <table border="1" data-bbox="435 625 1416 800"> <thead> <tr> <th data-bbox="435 625 927 686">Event</th> <th data-bbox="927 625 1416 686">Notification sent</th> </tr> </thead> <tbody> <tr> <td data-bbox="435 686 927 747">Virtual Service Started</td> <td data-bbox="927 686 1416 747">When a virtual service is started.</td> </tr> <tr> <td data-bbox="435 747 927 806">Virtual Service Stopped Manually</td> <td data-bbox="927 747 1416 806">When a virtual service is manually stopped.</td> </tr> </tbody> </table>		Event	Notification sent	Virtual Service Started	When a virtual service is started.	Virtual Service Stopped Manually	When a virtual service is manually stopped.		
Event	Notification sent									
Virtual Service Started	When a virtual service is started.									
Virtual Service Stopped Manually	When a virtual service is manually stopped.									
Virtual Service Errors	<p>Webhook sends notifications about the errors that occur in the virtual services.</p> <p>Following are the subsets of the virtual service error events:</p> <table border="1" data-bbox="435 945 1416 1211"> <thead> <tr> <th data-bbox="435 945 927 1005">Event</th> <th data-bbox="927 945 1416 1005">Notification sent</th> </tr> </thead> <tbody> <tr> <td data-bbox="435 1005 927 1066">Virtual Service Launch Failed</td> <td data-bbox="927 1005 1416 1066">When a virtual service launch fails.</td> </tr> <tr> <td data-bbox="435 1066 927 1127">Virtual Service Stopped with Error</td> <td data-bbox="927 1066 1416 1127">When a virtual service is stopped with an error.</td> </tr> <tr> <td data-bbox="435 1127 927 1211">Virtual Service Stopped Automatically</td> <td data-bbox="927 1127 1416 1211">When a virtual service is automatically stopped.</td> </tr> </tbody> </table>		Event	Notification sent	Virtual Service Launch Failed	When a virtual service launch fails.	Virtual Service Stopped with Error	When a virtual service is stopped with an error.	Virtual Service Stopped Automatically	When a virtual service is automatically stopped.
Event	Notification sent									
Virtual Service Launch Failed	When a virtual service launch fails.									
Virtual Service Stopped with Error	When a virtual service is stopped with an error.									
Virtual Service Stopped Automatically	When a virtual service is automatically stopped.									

Table 58. License Events

Event	Notification sent
License Failure	When a valid license is not obtained.

Chapter 8. Troubleshooting

This guide describes how to analyze and resolve some of the common problems that you might encounter while you work with HCL DevOps Test Hub (Test Hub).

Known issues are documented in the Release document and the link is in the Release Notes under the Known issues section. See [Defect fixes and known issues on page 9](#).

You can contact HCL Support if you are unable to troubleshoot the problem. Gather all the required background information and provide the details to HCL Support for investigation. For more information, see [HCL Customer Support](#).

Troubleshooting issues

You can find information about the issues or problems that you might encounter while working with HCL DevOps Test Hub (Test Hub). Details about issues, their causes and the resolutions that you can apply to fix the issues are described.

The troubleshooting issues are presented to you in the following tables based on where or when you might encounter these issues on Test Hub.

- [Table 59: Troubleshooting issues: installation on page 1077](#)
- [Table 60: Troubleshooting issues: server administration on page 1080](#)
- [Table 61: Troubleshooting issues: remote Docker host on page 1080](#)
- [Table 62: Troubleshooting issues: configuring test runs on page 1082](#)
- [Table 63: Troubleshooting issues: test or stub runs on page 1083](#)
- [Table 65: Troubleshooting issues: Jira and Xray integrations on page 1087](#)

Table 59. Troubleshooting issues: installation

Problem	Cause	Solution
On Ubuntu, when you are installing Test Hub and you encounter errors in the scripts that are running.	At times, scripts might not appear to be running due to any of the following reasons: <ul style="list-style-type: none">• Slow connection speeds.• Insufficient CPU, memory, or disk resources.• A firewall that was configured incorrectly is already enabled.	You can complete any of the following tasks: <ul style="list-style-type: none">• To identify the issue, you can perform a diagnostic check by running the following command:<pre>journalctl -u k3s</pre>This command displays the log that you can use to check for the problem.

Table 59. Troubleshooting issues: installation (continued)

Problem	Cause	Solution
		<ul style="list-style-type: none"> Run the following command to see which pods are running and which pods are not running: <pre data-bbox="1117 533 1421 569">kubectġ get pods -A</pre> Run the following command to get details about a specific pod: <pre data-bbox="1117 747 1421 842">kubectġ describe pod -n <namespace> <pod name></pre> Follow the on-screen instructions to resolve the errors. Some issues can be solved by re-running the following script: <pre data-bbox="1117 1087 1421 1123">sudo ./k3s-init.sh</pre>
On Ubuntu, DNS is not working as expected.		<p>The DNS configuration that is used by the cluster can be displayed by using the following command:</p> <pre data-bbox="1040 1314 1421 1440">kubectġ get cm -n kube-system coredns -ojsonpath="{.data.Corefile}"</pre> <p>The forward setting displays the <i>nameservers</i> that are used. For example, you might see the following in the <i>corefile</i>:</p> <pre data-bbox="1040 1646 1421 1837">.:53 { : forward . 8.8.8.8 9.9.9.9 : }</pre>

Table 59. Troubleshooting issues: installation (continued)


Problem	Cause	Solution
		<p>A script (<code>ubuntu-set-dns.sh</code>) is supplied for managing these values.</p> <p>For example, to set the DNS values for the values shown in the previous example:</p> <pre>sudo ./ubuntu-set-dns.sh --server 8.8.8.8 --server 9.9.9.9</pre> <p> Note: If you do not use <code>sudo</code> in the command, the script runs but the configuration might be lost if the cluster is restarted.</p> <p>To learn more about the behavior of the script, run the following command:</p> <pre>sudo ./ubuntu-set-dns.sh --help</pre>
<p>When running <code>helm install</code> the created pods keep crashing, and the logs contain: <code>ACCESS_REFUSED</code> when trying to connect to RabbitMQ</p>	<p>In some instances, the RabbitMQ password is not automatically setup correctly.</p>	<p>Manually apply the necessary password:</p> <pre>kubectl exec -n <namespace> <release-name>-rabbitmq-0 -- rabbitmqctl change_password user \ "\$(kubectl get secret -n <namespace> <release-name>-rabbitmq -o jsonpath='{.data.rabbitmq-p assword}' base64 --decode)"</pre>

Table 60. Troubleshooting issues: server administration

Problem	Cause	Solution
When a user is assigned an additional role, the change in the permissions is not observed in the browser.		You must log out of the session and log in again for the changed role to take effect.
<p>You see the following message displayed on Test Hub:</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px 0;"> <p>You can't request to join a project that has no owners</p> </div>	You requested to join a project that no longer has an owner. Orphaned projects occur when the project owners are deleted. This can occur, for example, when the person leaves the organization.	Ask an administrator to take ownership of the project, and then add you as a member.

Table 61. Troubleshooting issues: remote Docker host

Problem	Cause	Solution
You encounter errors when you attempt to register a remote Docker host.	The computer on which you set up the remote Docker is not running or accessible.	You can issue a ping command by using the hostname or IP address of the remote Docker host computer to verify that the remote host is running and accessible.
	The Common Name or Fully Qualified Domain Name in the Certificate Authority does not match the hostname you specified as the remote Docker host.	Verify by running the following command: <code>\$ openssl x509 -in ca.pem -text</code> and check the value of the CN field returned. If the names match, this is not the problem. If the names do not match, change the name in the certificate.
	The argument <code>-traditional</code> was not specified when you generated the RSA key.	Ensure that you use the <code>-traditional</code> argument when you generate the RSA key.
	Connection with the remote Docker host computer fails.	Perform the following steps:

Table 61. Troubleshooting issues: remote Docker host (continued)

Problem	Cause	Solution
		<ol style="list-style-type: none"> 1. Start the Docker daemon manually in a terminal window on the remote Docker host computer. 2. Click the Test Connection in the New docker host dialog on the Agents and Intercepts page. 3. Look for any messages that might indicate the root cause of the problem such as <code>bad certificate</code>. <p>Attempt to access the remote Docker host computer by using the same CA certificate, Client certificate, and Client key from a terminal window on any computer that is on the same network as Test Hub by running the following command:</p> <pre style="background-color: #f0f0f0; padding: 5px;">\$ docker --tlsverify \ --tlscacert=ca.pem \ --tlscert=cert.pem \ --tlskey=key.pem \ -H=\$HOST:2376 version</pre> <p>Where <i>HOST</i> is the hostname of the remote Docker host computer.</p> <p>The output from this command must be clean information about the version of Docker running on the remote host with no errors or exceptions.</p>

Table 62. Troubleshooting issues: configuring test runs

Problem	Cause	Solution
<p>When you configure a run of a schedule that matches the following conditions:</p> <ul style="list-style-type: none"> • The schedule has two user groups configured to run on static agents when the schedule was created in Test Performance 10.1. • One of the user groups is disabled and the asset is committed to the remote repository. <p>Both the static agents are displayed as available for the test run in the Location tab of the Execute test asset dialog when only one agent that is configured for the user group must be available.</p>	<p>The cause might be because of the following reasons:</p> <ul style="list-style-type: none"> • The schedule was created in Test Performance 10.1. • The user group that is disabled is not removed or deleted from the test resources. • The agent configured on the disabled user group is already added as an agent to the server project and is available for selection. 	<p>To resolve the problem, select from either of the following methods:</p> <ul style="list-style-type: none"> • By using Test Performance 10.1.1. <p>Perform the following steps:</p> <ol style="list-style-type: none"> 1. Open the schedule in Test Performance 10.1.1. 2. Save the schedule and the project. 3. Commit your test asset to the remote repository. 4. Proceed to configure a run for the schedule on Test Hub 10.1.1. <ul style="list-style-type: none"> • By using Test Performance 10.1. <p>Perform the following steps:</p> <ol style="list-style-type: none"> 1. Select the disabled user group. 2. Click Remove. 3. Save the schedule and the project. 4. Commit your test asset to the remote repository. 5. Proceed to configure a run for the schedule on Test Hub 10.1.1.
<p>You have added a remote repository to your project that contains the test assets or resources of the following types:</p>	<p>This problem occurs if the server extension is not enabled. Although the extension was enabled at the time of installation of Test Hub, it was dis-</p>	<p>You must verify if the server extension is enabled and running by running the following command: <code>kubectl get pod -n devops-system</code>, where <code>de-</code></p>

Table 62. Troubleshooting issues: configuring test runs (continued)

Problem	Cause	Solution
<ul style="list-style-type: none"> • Postman • JMeter • JUnit <p>The test assets or resources are not displayed on the Execution page for you to select the asset for a run.</p>	<p>abled subsequently by the server administrator.</p>	<p><i>vops-system</i> is the namespace that you created to install Test Hub. The server extensions that are running are displayed.</p> <p>If the server extension that you want is not running implying that the server extension is not enabled. You must enable the server extension. Contact the server administrator to enable the server extension.</p>

Table 63. Troubleshooting issues: test or stub runs



Problem	Cause	Solution
<p>You encounter any of the following issues:</p> <ul style="list-style-type: none"> • When many tests are run simultaneously on the default cluster location and you observe the following issues: <ul style="list-style-type: none"> ◦ Out of memory errors. ◦ Observe that the test runs are slow with a high CPU usage. ◦ The Kubernetes pods are getting evicted. • When you run an AFT suite that contains multiple Web UI tests and you observe the following issues: <ul style="list-style-type: none"> ◦ Error stating that the browser might not be installed or the brows- 	<p>The issue is seen when any of the following events occur:</p> <ul style="list-style-type: none"> • Many tests are run in parallel. • The memory that is used by the tests during the test run exceeds the allocated default memory of 1 GB. • The default memory of the container is not adequate for the test run. • Pods are evicted due to low node memory. 	<p>To resolve the problem, you can increase the resource allocation for test runs by using the arguments listed in Table 64: Increasing resource allocation on page 1085.</p> <p>You can enter arguments in the Additional configuration options field in the Advanced settings panel of the Execute test asset dialog when configuring a test run.</p> <p> Important: The memory settings that you configure for a test run is persisted for the test when ever you run it. You must use this setting judiciously. Configuring all tests for an increased memory limit might affect subsequent test runs or cause oth-</p>

Table 63. Troubleshooting issues: test or stub runs (continued)

Problem	Cause	Solution
<p>er version is unsupported.</p> <ul style="list-style-type: none"> ◦ Error stating multiple random time-outs or an internal error. • When you run VU schedules that contain multiple Web UI test or large number of virtual users or a combination of Web UI, performance, and API tests, and you observe the following issue: <ul style="list-style-type: none"> ◦ The test run hangs and the inadequacy of memory can be inferred as the cause in the logs. 		<p> er memory issues when tests run simultaneously.</p> <p>In addition, in the JVM Arguments field under the Advanced settings you can set the maximum heap size for the test run time. For example, adding the JVM argument -Xmx3g sets the maximum heap size to 3Gi.</p>
<p>You are not able to run Web UI tests on a remote agent that runs on Linux because the browser fails to launch.</p>	<p>Browser launch is prevented when the remote agent runs as <i>root</i> or <i>sudo</i>.</p>	<p>To resolve this issue, you must perform the following steps after you have installed the remote agent:</p> <ol style="list-style-type: none"> 1. Stop the agent. 2. Start the agent as the logged in user by performing the following steps: <ol style="list-style-type: none"> a. Open a terminal on Linux. b. Enter the following commands: <pre data-bbox="1182 1556 1419 1755">cd /opt/HCL/DevOps-Test/Majordomo/ sudo ./MDStop.sh ./MDStart.sh</pre>



Note: You can refer to the [Kubernetes documentation](#) for information about the different units that can be used for resources in the **Additional configuration option** fields.

Table 64. Increasing resource allocation


Requirement	Configuration option name	Default value, if no value is set	An example value	Result of using the example value
Specifying the memory limit of the <i>init container</i> .	<code>init.resource.memory.limit</code>	<i>1024Mi</i>	<i>2048Mi</i>	Increases the memory limit of the <i>init container</i> from the default value to <i>2048Mi</i> .
Configuring a larger memory request for the <i>init container</i> to avoid pod eviction.	<code>init.resource.memory.request</code>	<i>64Mi</i>	<i>1024Mi</i>	Increases the initial memory request for the <i>init container</i> from the default value to <i>1024Mi</i> .
Specifying the cpu request for the <i>init container</i> .	<code>init.resource.cpu.request</code>	<i>50m</i>	<i>60m</i>	Increases the cpu request for the <i>init container</i> from the default value <i>60m</i> .
Specifying the memory limit of the container used for the test run.  Note: If the memory limit you set is more than the default limit, you must run the following command to increase the allotted lim-	<code>resource.memory.limit</code>	The larger of <i>3Gi</i> or <i>maximum heap size + 1Gi</i>	<i>4Gi</i>	Changes the memory limit of the main container from the default value to <i>4Gi</i> .

Table 64. Increasing resource allocation

(continued)


Requirement	Configuration option name	Default value, if no value is set	An example value	Result of using the example value
<p> it to 8 GB or any other value above the default value of 3 GB, at the time of installing the server or any-time later:</p> <pre>--set execution .template. resources. limits.mem ory=8Gi</pre> <p>You must be a server administrator to run this command.</p>				
Specifying the memory request for the container used by the test run.	<code>resource.memory.request</code>	<i>1024Mi</i>	<i>2048Mi</i>	Increases the memory request for the main container from the default value to <i>2048Mi</i> .
Specifying the cpu request for the main container used by the test run.	<code>resource.cpu.request</code>	<i>50m</i>	<i>70m</i>	Increases the cpu request for the main container from the default value to <i>70m</i> .

Table 65. Troubleshooting issues: Jira and Xray integrations

Problem	Cause	Solution
You are not able to create Jira work items in Jira. You are also not able to view the created work items in Jira that you created in HCL OneTest™ Server 10.5.4 or earlier.	The cause can be due to the change in the authentication method used to connect to Jira in Test Hub 11.0.0 or later.	Reconfigure Jira as a change management system in Test Hub 11.0.0 or later. See Configuration of Atlassian Jira as a change management system on page 892 .
You are not able to export results to the Xray project. You are also not able to view the results exported to the Xray project that you created in HCL OneTest™ Server 10.5.4 or earlier.	The cause can be due to the change in the authentication method used to connect to Jira in Test Hub 11.0.0 or later.	Reconfigure the Xray project after you reconfigure Jira Test Hub 11.0.0 or later. See Integration with Xray test management for Jira on page 988 .

Security Considerations

This document describes the actions that you can take to ensure that your installation is secure, customize your security settings, and set up user access controls in HCL DevOps Test Hub (Test Hub).

- [Enabling secure communication between multiple applications on page mlxxxviii](#)
- [Ports, protocols, and services on page mlxxxviii](#)
- [Customizing your security settings on page mlxxxviii](#)
- [Setting up user roles and access on page mlxxxix](#)

Enabling secure communication between multiple applications

The majority of communications are sent over TLS to port 443 (see [Ports, protocols, and services on page mlxxxviii](#)). During the installation, an X.509 certificate is generated for the user provided DNS name, which is used to connect to the server. This certificate is self-signed and hence untrusted by other applications.

This self-signed certificate must be replaced by a certificate signed by a certificate authority trusted by your organization. For more information, see [X.509 Certificate User Authentication](#) in the Keycloak documentation.

For information about how the self-signed certificate was created, see the `ssl.sh` file in the `<install-directory>/prepare/` directory.

For information about importing a certificate authority trusted by your organization, see [Certificate authority: Importing and extending lists on page 156](#).

Ports, protocols, and services

TCP port 443 is used by the majority of communications with the server.

The port 7085 is the default port for communications with agents registered with Test Hub.

The ports starting from 7085, are used in pairs such as 7085 and 7086, and are allotted for the Schedule that is executed first. The next Schedule is allotted the next pair (7087,7088), and so on for the Schedules that are running simultaneously.

You must open the required ports in pairs for each of the Schedules that you want to run simultaneously.

Customizing your security settings

You can customize your security settings through user registration.

User registration

By default, users can sign up themselves with the server. In some environments, this self sign-up might be undesirable. It can be changed by switching off user registration. For more information, see [User Registration](#) in the Keycloak documentation.

By default, user email addresses are not verified. This verification must be enabled in production environments. For more information, see [Email settings on page 140](#).

Setting up user roles and access

You can manage user roles and access through single sign on (SSO) and administration only accounts.

Single sign-on

By default, Keycloak manages users and passwords locally. In production environments, it is normally appropriate to use single sign-on. For more information, see [LDAP user administration on page 141](#).

Administration only accounts

Users in the Administrator group can manage the team space where they can configure licenses and notifications and add a repository to store the System model. For more information, see [Team space overview on page 168](#).

Users in the Administrator group can also discover all projects stored on the server (including private ones) and assign themselves and others roles in those projects.

For this reason, users who use the server to perform both administration and non-administration tasks must have two different accounts, one for each purpose. For more information, see [Default user administration on page 139](#).

Notices

This document provides information about copyright, trademarks, terms and conditions for the product documentation.

© Copyright IBM Corporation 2000, 2016 / © Copyright HCL Technologies Limited 2016, 2024

This information was developed for products and services offered in the US.

HCL® may not offer the products, services, or features discussed in this document in other countries. Consult your local HCL® representative for information on the products and services currently available in your area. Any reference to an HCL® product, program, or service is not intended to state or imply that only that HCL® product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any HCL® intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-HCL® product, program, or service.

HCL® may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA
Attention: Office of the General Counsel

For license inquiries regarding double-byte character set (DBCS) information, contact the HCL® Intellectual Property Department in your country or send inquiries, in writing, to:

HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA
Attention: Office of the General Counsel

HCL TECHNOLOGIES LTD. PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. HCL® may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-HCL® websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this HCL® product and use of those websites is at your own risk.

HCL® may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

HCL

330 Potrero Ave.

Sunnyvale, CA 94085

USA

Attention: Office of the General Counsel

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by HCL® under terms of the HCL® Customer Agreement, HCL® International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-HCL® products was obtained from the suppliers of those products, their published announcements or other publicly available sources. HCL® has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-HCL® products. Questions on the capabilities of non-HCL® products should be addressed to the suppliers of those products.

Statements regarding the future direction or intent of HCL® are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to HCL®, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. HCL®, therefore, cannot guarantee or imply reliability,

serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. HCL® shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from HCL Ltd. Sample Programs.

© Copyright HCL Ltd. 2000, 2022.

Trademarks

HCL®, the HCL® logo, and hcl.com® are trademarks or registered trademarks of HCL Technologies Ltd., registered in many jurisdictions worldwide. Other product and service names might be trademarks of HCL® or other companies.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the HCL® website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of HCL®.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of HCL®.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

HCL® reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by HCL®, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

HCL® MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

A

- accessibility
 - Test Hub
 - disability 92
 - keyboard 92
- administration default security
 - Test Hub
 - 139
- administration ldap security
 - Test Hub
 - 141

C

- Change
 - Classification 243
 - Dataset 243
 - Encrypt 243
- Change Classification
 - Dataset 237
- Configure
 - Dataset 237
- Create
 - Classification 243

D

- Dataset 237
- datasets
 - encrypting 233
 - removing encryption 235
- Delete
 - Classification 243

E

- Edit
 - Classification 243

G

- getting started
 - guide 87
- Grant
 - access 243
 - Classification 243
- guide
 - getting started 87

M

- Manage encrypted dataset 243

O

- overview
 - server 87

R

- Remove
 - Classification 243
 - Dataset 243
 - Encrypt 243
- Revoke
 - access 243
 - Classification 243

S

- security considerations
 - overview mlxxxviii
- server
 - overview 87
- Set
 - Current row 237
 - Dataset 237

T

- Test Hub
 - accessibility 92
 - default security administration 139
 - disability 92
 - keyboard 92
 - ldap security administration 141
- tests
 - datasets 233, 235

V

- View
 - Dataset 237, 237
 - Encrypt 237