# HCL BigFix Remediate v11

# on AWS Marketplace

## User Guide

Version 1.0

BigFix

HCLSoftware

# List of Figures

# 1 Introduction

"HCL BigFix Remediate v11" is available on the AWS Marketplace.

Its delivery method is "CloudFormation Template", so it can be deployed through the AWS CloudFormation service (hereinafter simply referred to as "CloudFormation"). CloudFormation will first create a so-called **stack** of AWS resources (EC2 instances, VPC, subnets, security groups, etc.), and then install **BigFix Platform v11.0.2** (hereinafter simply referred to as "BigFix") on top of them.

The licensing model is **BYOL** (Bring Your Own License), which implies that customers will need to purchase a BigFix license via official HCL channels prior to deploying the product from the AWS Marketplace. The license must be for the HCL BigFix Remediate offering.

The overall cost of the solution must also consider the infrastructural costs charged by AWS.

The core BigFix infrastructure generated by CloudFormation is made up of five Windows Server 2022 Datacenter instances, and is organized as depicted in the orange frame of the following *Figure 1*, where the labels contained in each instance describe the BigFix components or middleware installed there, or the role played by the instance (for a more detailed description of all created resources, go to **Appendix A**):

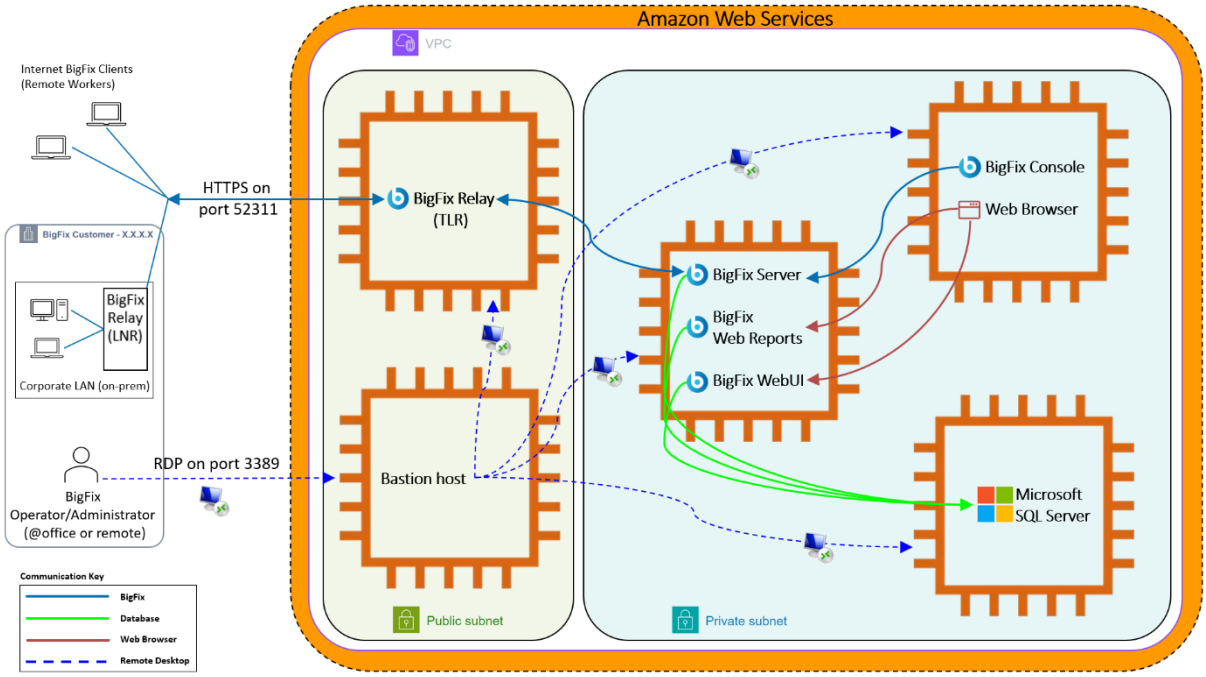


*Figure 1: BigFix core infrastructure generated by CloudFormation (orange frame).*

IMPORTANT NOTE: Once created via CloudFormation, the customer will maintain full control over the BigFix deployment and all AWS resources.

This initial BigFix infrastructure is suited to benefit from most of the "HCL BigFix Remediate" use cases and can be easily expanded with additional EC2 instances that would enable the deployment of HCL BigFix Insights, more Top-Level Relays, and a Plugin Portal so as to take full advantage of the "HCL BigFix Remediate" offering.

Information about system requirements and how to install and configure "HCL BigFix Insights" can be found at https://help.hcltechsw.com/bigfix/11.0/insights/index.html.

## 1.1 Terminology

### 1.1.1 Instance and EC2 Instance

Throughout this guide we will use "instance" and "EC2 instance" as equivalent expressions.

### 1.1.2 BigFix Components by Instance

These are the BigFix components installed on AWS by CloudFormation:

- On the BigFix Server instance:
  - BigFix Server ("Server" from now on)
  - BigFix Web Reports ("Web Reports" from now on)
  - BigFix WebUI ("WebUI" from now on)
- On the BigFix Console instance:
  - BigFix Console ("Console" from now on)
- On the BigFix Relay instance:
  - BigFix Relay ("Relay" from now on)

NOTE: All above instances will also run the BigFix Client component ("Client" from now on). This is taken for granted and not further specified from now on.

### 1.1.3 Top-Level Relay (TLR) and Leaf Node Relay (LNR)

Throughout this guide we will use the terms Top-Level Relay (or TLR) and Leaf Node Relay (or LNR) as follows:

**Top-Level Relay** (or simply **TLR**): the Relay installed on AWS by CloudFormation. It will play as parent for Clients and Relays installed by the customer.

**Leaf Node Relay** (or simply **LNR**): any on-prem Relay installed by the customer and connected to the TLR.

# 2 Getting Started

This section describes everything that needs to be in place or prepared before you start creating the stack through CloudFormation.

## 2.1 Prerequisites

### 2.1.1 BigFix Prerequisites

Before installing "HCL BigFix Remediate v11" from the AWS Marketplace, you must have purchased a license and obtained the related BigFix license authorization file ("**.BESLicenseAuthorization*") from your HCL Technical Sales Representative, or using your HCLSoftware License & Download Portal account. The license must strictly be for the HCL BigFix Remediate offering only.

IMPORTANT NOTE: Using a BigFix license that contains offerings other than "HCL BigFix Remediate" will cause the installation of BigFix to fail. For further details, check the **License Check Failed** troubleshooting section.

Other prerequisites concern the BigFix administrator, who should provide the following:

- A site admin private key password, which will be used to encrypt the site admin private key file (*license.pvk*) during the installation of BigFix. This password will be always necessary to work with the BigFix Administration Tool. Be aware that in case this password is forgotten, there is no way to recover it, and BigFix will need to be reinstalled from scratch using a new license.

- A username and a password for the BigFix master operator to be created during the installation of BigFix and who will serve as BigFix administrative user.

### 2.1.2 AWS Prerequisites

Here is a list of AWS prerequisites:

- An active AWS account.

- A chosen AWS Region where all stack resources will be created.

- A IAM service role that CloudFormation can assume and use for all operations performed on the stack, in the chosen AWS Region. For further details, go to **Appendix B**.

  NOTE: The IAM service role is not necessary in case the IAM user who launches CloudFormation has the permissions to perform all stack operations in the chosen AWS Region.

- The availability of an Amazon S3 bucket where to store the BigFix license authorization file ("**.BESLicenseAuthorization*") so that it can be automatically downloaded and used during the creation of the stack and the installation of BigFix.

- At least one key pair available in the chosen AWS Region. Key pairs allow connecting to EC2 instances securely. If you plan to use a different key pair for each EC2 instance of the stack, ensure you have 5 different key pairs available in the chosen AWS Region. Whatever the number, also ensure you have access to all the key pairs that you plan to use because the unavailability of a key pair specified at CloudFormation launch time will prevent the access to the related EC2 instances.

- An IPv4 CIDR block for the creation of one VPC, and two related subnets.

- Possibility to create 2 Elastic IP addresses without exceeding the Elastic IP address quota for the chosen AWS Region.

### 2.1.3 Microsoft SQL Server Prerequisites

A database administrator should provide a login name and a password for an administrative user to be created on Microsoft SQL Server. These credentials, which are asked at CloudFormation launch time, will be then used by the Server, Web Reports and WebUI components to connect to the database through SQL Server authentication.

## 2.2 Capacity Planning

At CloudFormation launch time, you will be requested to input an instance type and a disk size for each of the following instances:

- BigFix Server instance (running Server, Web Reports and WebUI).
- BigFix Console instance (running the Console).
- BigFix Relay instance (running the Relay).
- Microsoft SQL Server instance.

For a list of recommended values, go to **Appendix C**.

## 2.3 Upload of the BigFix License to an Amazon S3 Bucket

Upload the BigFix license authorization file ("**.BESLicenseAuthorization*") to an Amazon S3 bucket that can then be accessed by CloudFormation for download purposes during the creation of the stack.

# 3 Install BigFix from the AWS Marketplace

From the AWS Console, follow these steps:

- Launch the "AWS Marketplace" service and go to "Discover products".
- Search for "HCL BigFix Remediate v11", then click on it.
- Click on "Continue to Subscribe", which will bring you to the "Subscribe to this software" page.
- Review the "Terms and Conditions" section – including pricing information and the seller's EULA – then accept "HCL Software Offer" terms if you want to continue.
- Wait for the terms acceptance request to be processed until completion (upon completion you will see the value of the current date appear in the "Effective date" column).
- Click on "Continue to Configuration", which will bring you to the "Configure this software" page.
- Select the following values:
    - "Fulfillment option" = "HCL BigFix Remediate".
    - "Software version" = "HCL BigFix Remediate 11.0.2 (May 30, 2024)".
    - In the "Region" selector, choose an AWS Region. This will be the AWS Region where all stack resources will be created.

      IMPORTANT NOTE: The final cost of the infrastructure may vary depending on the AWS Region chosen.

- Click on "Continue to Launch", which will bring you to the "Launch this software" page.
- Click on the "Usage instructions" to get the latest version of this user guide.
- In "Choose Action", select "Launch CloudFormation" and the click on the "Launch" button.

In the next paragraphs, CloudFormation launch options specifically relevant to BigFix will be listed and described, while the more general ones will be left to the user to choose according to corporate policies.

## 3.1 Step 1 – "Create stack"

→ In the "Prerequisite - Prepare template" section, leave the pre-selected "Choose an existing template" option.

→ In the "Specify template " section, leave the pre-selected "Amazon S3 URL" option.

→ Click "Next".

## 3.2 Step 2 – "Specify stack details"

→ Provide a stack name of your choice. Note that the provided stack name will be prepended to the names of some of the stack resources that will be created by CloudFormation.

→ In the "Parameters" section, provide the following information:

BigFix License

- **S3 bucket name**: *name of the S3 bucket where the BigFix license authorization file (*.BESLicenseAuthorization) is stored.*

- **BigFix license authorization file**: *name of the BigFix license authorization file (*.BESLicenseAuthorization) stored on the S3 bucket. **IMPORTANT NOTE**: Make sure the specified BigFix license authorization file contains only "Remediate" allowances otherwise stack resources will be created, but BigFix will not be installed.*

- **Site admin private key password**: *password to encrypt the site admin private key file (license.pvk). This password will be always necessary to work with the BigFix Administration Tool. Be aware that in case this password is forgotten, there is no way to recover it, and BigFix will need to be reinstalled from scratch using a new license.*

- **Confirm site admin private key password**

BigFix Server EC2 instance

- **BigFix Server instance type**: *type of the instance hosting the BigFix Server, Web Reports and WebUI. For more information about how to choose a suitable value, refer to appendix **C.1**. Note that this value can be changed later in case of new or updated requirements.*

- **BigFix Server disk type**: *disk type of the instance hosting the BigFix Server, Web Reports and WebUI. More information can be found at https://docs.aws.amazon.com/ebs/latest/userguide/ebs-volume-types.html. Note that this value can be changed later in case of new or updated requirements.*

- **BigFix Server disk size (GB)**: *disk size of the instance hosting the BigFix Server, Web Reports and WebUI. For more information about how to choose a suitable value, refer to appendix **C.1**. Note that this value can be changed later in case of new or updated requirements.*

- **BigFix Server hostname**: *computer name of the instance hosting the BigFix Server. Must be from 6 to 15 characters long (ASCII characters only). This name will be permanently associated with the BigFix license and to the BigFix deployment masthead. **IMPORTANT NOTE**: Any BigFix Clients that will be installed on the public or private AWS subnet must be able to resolve this provided hostname to the private IP address of the instance.*

- **Key pair for the BigFix Server instance**: *a key pair name available to the chosen AWS Region for securely connecting to the BigFix Server instance.*

- **BigFix master operator username**: *name of the BigFix master operator who will serve as BigFix administrative user.*

- **BigFix master operator password**: *password of the BigFix master operator. Minimum 8 characters long, with at least one uppercase letter, one lowercase letter, and one digit.*

- **Confirm BigFix master operator password**

BigFix Relay EC2 instance

- **BigFix Relay instance type**: *type of the instance hosting the BigFix Relay. For more information about how to choose a suitable value, refer to appendix **C.2**. Note that this value can be changed later in case of new or updated requirements.*

- **BigFix Relay disk type**: *disk type of the instance hosting the BigFix Relay. More information can be found at https://docs.aws.amazon.com/ebs/latest/userguide/ebs-volume-types.html. Note that this value can be changed later in case of new or updated requirements.*

- **BigFix Relay disk size (GB)**: *disk size of the instance hosting the BigFix Relay. For more information about how to choose a suitable value, refer to appendix **C.2**. Note that this value can be changed later in case of new or updated requirements.*

- **BigFix Relay hostname**: *computer name of the instance hosting the BigFix Relay. Must be from 6 to 15 characters long (ASCII characters only). IMPORTANT NOTE: BigFix Clients and Relays (the latter also referred to as Leaf Node Relays) that will connect to this BigFix Relay (TLR) must be able to resolve this provided hostname to the public Elastic IP address of the instance.*

- **Key pair for the BigFix Relay instance**: *a key pair name available to the chosen AWS Region for securely connecting to the BigFix Relay instance.*

BigFix Console EC2 instance

- **BigFix Console instance type**: *type of the instance hosting the BigFix Console. For more information about how to choose a suitable value, refer to appendix **C.3**. Note that this value can be changed later in case of new or updated requirements.*

- **BigFix Console disk type**: *disk type of the instance hosting the BigFix Console. More information can be found at https://docs.aws.amazon.com/ebs/latest/userguide/ebs-volume-types.html. Note that this value can be changed later in case of new or updated requirements.*

- **BigFix Console disk size (GB)**: *disk size of the instance hosting the BigFix Console. For more information about how to choose a suitable value, refer to appendix **C.3**. Note that this value can be changed later in case of new or updated requirements.*

- **Key pair for the BigFix Console instance**: *a key pair name available to the chosen AWS Region for securely connecting to the BigFix Console instance.*

Microsoft SQL Server EC2 instance

- **Microsoft SQL Server instance type**: *type of the instance hosting Microsoft SQL Server. For more information about how to choose a suitable value, refer to appendix **C.4**. Note that this value can be changed in the future in case of new or updated requirements.*

- **Microsoft SQL Server disk type**: *disk type of the instance hosting Microsoft SQL Server. More information can be found at https://docs.aws.amazon.com/ebs/latest/userguide/ebs-volume-types.html. Note that this value can be changed later in case of new or updated requirements.*

- **Microsoft SQL Server disk size (GB)**: *disk size of the instance hosting Microsoft SQL Server. For more information about how to choose a suitable value, refer to appendix **C.4**. Note that this value can be changed in the future in case of new or updated requirements.*

- **Key pair for the Microsoft SQL Server instance**: *a key pair name available to the chosen AWS Region for securely connecting to the Microsoft SQL Server instance.*

- **Microsoft SQL Server administrative login name**: *new administrative login name for SQL Server authentication (the value 'sa' is not allowed). It will be used by the BigFix Server to connect to the database.*

- **Microsoft SQL Server administrative login password**: *administrative login password.*

- **Confirm Microsoft SQL Server administrative login password**

Bastion EC2 instance

- **Key pair for the bastion instance**: *a key pair name available to the chosen AWS Region for securely connecting to the bastion instance.*

- **Inbound source for RDP access to the bastion instance**: *CIDR block or IP address from which the bastion instance can be accessed via RDP. Must be specified using the CIDR subnet mask notation (x.x.x.x/x). For example, a value of "x.x.x.x/32" would restrict inbound RDP access to a single IP address, while a value of "0.0.0.0/0" would allow inbound RDP access from any IP address.*

Network configuration (VPC and subnets creation)

- **VPC – IPv4 CIDR block**: *the chosen IPv4 address range for the VPC. Specify the IPv4 address range as a CIDR block. A CIDR block size must be between a /16 netmask and /28 netmask; for example, 192.168.186.0/24.*

- **Public subnet – IPv4 subnet CIDR block**: *this is the subnet of the BigFix Relay and of the bastion instances. The IPv4 CIDR block of the subnet must lie within the IPv4 CIDR block of the VPC; for example 192.168.186.0/25.*

- **Private subnet – IPv4 subnet CIDR block**: *this is the subnet of the BigFix Server, BigFix Console and Microsoft SQL Server instances. The IPv4 CIDR block of the subnet must lie within the IPv4 CIDR block of the VPC; for example 192.168.186.128/25.*

### 3.3 Step 3 – "Configure stack options"

→ In the "Permissions - optional" section, select a IAM service role that CloudFormation can assume and use for all operations performed on the stack, in the chosen AWS Region. For further details, go to **Appendix B**.

NOTE: The IAM service role is not necessary in case the IAM user who launches CloudFormation has the permissions to perform all stack operations in the chosen AWS Region.

→ Choose all the other options according to corporate policies.

### 3.4 Step 4 – "Review and create"

→ Review all inputs, then click on "Submit" to launch the creation of the stack.

NOTE: Before submitting, user is requested to acknowledge that CloudFormation might create IAM resources with custom names. This is expected because the stack resources will include a IAM role that would allow CloudFormation to access the S3 bucket where the BigFix license authorization file ("**.BESLicenseAuthorization*") is located.

### 3.5 Stack Creation and Background Tasks Completion

Wait until the new stack appears in CREATE_COMPLETE status in the "Stacks" area on the leftmost side of the AWS Console.

From that moment, wait an additional 30 minutes to allow background tasks to complete the deployment and configuration of all BigFix components. During this phase, do not access or edit any stack resources. Once the 30 minutes have passed, move on to the health check phase.

# 4 Health Check

## 4.1 Health Check Preparation Steps

### 4.1.1 Decrypt Instance Passwords

The availability of the decrypted passwords is key for subsequent RDP access to the instances.

On the AWS Console, from the "Resources" tab of the stack, do the following for each EC2 instance:

- Click on the "Physical ID" link, which will open the "Instances" page.
- Select the instance and click on "Connect".
- On the "RDP client" tab, optionally click on "Download remote desktop file" to get an "*xxxx.rdp*" file that can allow a quicker RDP access to the instance.
- Still on the "RDP client" tab, click on "Get password".
- Click on the "Upload private key file" button and then, in the "File upload" dialog box, select the key pair file associated with the selected instance.
- Click on "Decrypt password".
- Take note of the decrypted password, which will allow RDP connection as "Administrator" to the selected instance.

### 4.1.2 Open an RDP Connection to the Bastion Instance

Do one of the following to open an RDP connection to the bastion instance from your computer:

- If you previously downloaded it, double click on the "*xxxx.rdp*" file of the bastion instance and input related decrypted password.

- On the AWS Console, from the "Outputs" tab of the stack, take note of the public IP address of the bastion instance (key name "BastionPublicIP"). Then, use it along with the related decrypted password to open an RDP connection as "Administrator" from your computer.

NOTE: The public IP address of the bastion instance is dynamic, so it can change over time (this can typically happen upon system reboot). Should this happen, its current value can always be found from the AWS Console, by browsing the running instances via EC2 service.

#### 4.1.3 Open RDP Connections from Bastion to BigFix Console and BigFix Server Instances

Do one of the following to open RDP connections from the bastion instance to the BigFix Console and BigFix Server ones:

- If you previously downloaded them, copy the "*xxxx.rdp*" files of the BigFix Console and of the BigFix Server instances to the bastion instance, then double click on each of them and input related decrypted password.
- On the AWS Console, from the "Outputs" tab of the stack, take note of the private IP addresses of the BigFix Console and of the BigFix Server instances (key names "BigFixConsolePrivateIP" and "BigFixServerPrivateIP" respectively). Then, from the bastion instance, use them along with the related decrypted password to open RDP connections as "Administrator".

#### 4.1.4 (Optional) Open RDP Connections from Bastion to BigFix Relay and Microsoft SQL Server Instances

For the next health check phase, it is not needed to have RDP connections open to the BigFix Relay and Microsoft SQL Server Instances. However, if necessary, the same RDP connection options as for BigFix Console and BigFix Server instances apply.

### 4.2 Health Check Execution

#### 4.2.1 Health Check of the Server

From the BigFix Console instance, run the following verification steps:

1. Launch the Console (use the link on the Desktop), input BigFix master operator username and password as specified at CloudFormation launch time, then wait for the Console to open.
2. You should be seeing 3 online computers, representing the following:
    a. The Server computer, named as specified at CloudFormation launch time.
    b. The Relay computer, named as specified at CloudFormation launch time.
    c. The Console computer, with name automatically generated by AWS and starting with 'EC2'.
3. Select all 3 computers, open the context menu (right click), and do "Send Refresh". The "Last Report Time" column should be soon displaying an up-to-date value for all 3 computers.
4. Go to "Tools" > "Take Custom Action…", select all 3 computers in the "Target" tab, then go to the "Action Script" tab and replace the "`// Enter your action script here`" string with "`relay select`", and then click "OK". Action status should soon become "Completed" for all computers.
5. On the "All Content" navigation tree on the left, open "Dashboards" > "BES Support" > "Deployment Health Checks" and verify the status of all included health checks.
6. Still on the "All Content" navigation tree, open "Dashboards" > "BES Support" > "License Overview", and check that the "Remediate" entitlements show up as "VALID", with correct "Quantity", "Type", and "Expiration Date".

### 4.2.2 Health Check of the WebUI

From the BigFix Console instance, run the following verification steps:

1. Connect to the WebUI via the browser (the URL to enter in the browser address bar can be found on the AWS Console, "Outputs" tab of the stack, key name "WebUIURL").
2. Ignore the message about the connection not being private, and continue.

   NOTE: The browser presents the message about the connection not being private because WebUI is configured to use HTTPS by default when it gets installed, and creates its own SSL certificate during the installation. Information about how to configure WebUI to use a custom SSL certificate can be found at https://help.hcltechsw.com/bigfix/11.0/webui/WebUI/Admin_Guide/c_ssl_cert_configuration.html.

3. Log in to WebUI with BigFix master operator username and password as specified at CloudFormation launch time (they are the same credentials already used to open the Console).
4. Go to "Devices", select all 3 computers, and do "Administration" > "Send Client Refresh"
5. Refresh the "Devices" page, and check that the "Last Report Time" column shows an up-to-date value for all 3 computers.
6. Log out.

### 4.2.3 Health Check of the Web Reports

From the BigFix Server instance, run the following verification steps:

1. From the Windows "Start" menu, launch the "BigFix Diagnostics Tool". Verify that all listed tests are green-ticked. In case of tests showing the error icon, do the following:
   a. Look at the text area on the bottom of the diagnostics GUI, which shows details about any possible failed or ignored tests.
   b. Click on the question mark icon on the right of a failed test to go to the related on-line documentation.

   NOTE: The last test (namely "Checking that TCP/IP is enabled on SQL server") will be ignored as it doesn't apply to a Server using a remote database. This is correct.

2. Connect to Web Reports via the browser. The URL to enter in the browser address bar is https://localhost:8083/webreports.
3. Ignore the message about the connection not being private, and continue.

   NOTE: The browser presents the message about the connection not being private because Web Reports is configured to use HTTPS by default when it gets installed, and creates its own SSL certificate during the installation. Information about how to configure Web Reports to use a custom SSL certificate can be found at https://help.hcltechsw.com/bigfix/11.0/platform/Platform/Web_Reports/c_web_reports_https_settings.html.

4. Provide the required information to define an initial administrative user.
5. Go to "Report List", then click on "Computer Properties List".
6. Check that you see all 3 computers. Then, click on one of them and check the related information in the next page.
7. Log out.

### 4.2.4 Health Check of the BigFix Administration Tool

From the BigFix Server instance, run the following verification steps:

1. From Windows "Start", go to "BigFix" > "BigFix Administration Tool".
2. Click on the "Browse" button, and then in the "Site Admin Signing Key" dialog, select file "*C:\BESInstallers\license\license.pvk*".
3. Click "OK", and then input the site admin private key password as specified at CloudFormation launch time.
4. Check that the "BigFix Administration Tool" GUI opens.
5. In the "Masthead Management" tab, click on the "Export Masthead" button, specify a path and a filename, then click "Save".
6. Check that the masthead file is created in the specified path. The file may now be deleted as this was done only as a test.
7. Click "Cancel" to close the "BigFix Administration Tool" GUI.

# 5 Troubleshooting

If the health check phase highlights any unexpected behavior, there are logs that can be consulted to understand what may have gone wrong during installation or why a functional health check scenario does not work.

## 5.1 BigFix Installation Issues

On all BigFix instances, installation logs are in the "*C:\BESInstallers*" folder, and named as follows:

- **start.log** (all BigFix instances)

  NOTE: As long as "*start.log*" ends up with all `exit code 0` messages, you can safely ignore any possible previous messages like `Unable to connect to the remote server` and `InvalidOperation`, which are due to the Server being not yet ready during the first attempts to download the BigFix deployment masthead from it.

- **client.log** (all BigFix instances)
- **server.log** (BigFix Server instance)
- **webuiinstall.log** (BigFix Server instance)
- **setup.log** (BigFix Server instance)
- **relay.log** (BigFix Relay instance)

### 5.1.1 License Check Failed

If BigFix was not installed on the BigFix Server instance, and file "*C:\BESInstallers\start.log*" contains the message:

```
License check failed - exiting
```

it means that the specified BigFix license authorization file ("**.BESLicenseAuthorization*") contained at least one allowance other than "Remediate". In this case, a BigFix installation failure is expected because BigFix on AWS Marketplace supports only the "HCL BigFix Remediate" offering.

Suggested action: delete the stack, and then create a new one specifying a BigFix license authorization file ("**.BESLicenseAuthorization*") that contains only "Remediate" allowances.

## 5.2 BigFix Functional Issues

If functional issues are encountered during the health check phase, you can consult BigFix logs to obtain more information about the root cause of the issue.

### 5.2.1 Server Log

On the BigFix Server instance, check the following log file:

*C:\Program Files (x86)\BigFix Enterprise\BES Server**BESRelay.log***

You can possibly make it verbose via "BES Support" fixlet "Enable Server verbose log" (ID 4595).

### 5.2.2 Web Reports Log

You can enable Web Reports log via "BES Support" fixlet "Enable Web Reports Server log" (ID 4591).

### 5.2.3 WebUI Logs

On the BigFix Server instance, check the following log files:

*C:\Program Files (x86)\BigFix Enterprise\BES WebUI**service-wrapper.log***

*C:\Program Files (x86)\BigFix Enterprise\BES WebUI\WebUI\logs**<webuiapp>.log***

### 5.2.4 Relay Log

On the BigFix Relay instance, check the following log file:

*C:\Program Files (x86)\BigFix Enterprise\BES Relay**logfile.txt***

You can make it more verbose via "BES Support" fixlet "Enable Relay verbose log" (ID 4776).

### 5.2.5 Client Log

On all BigFix instances, check the following log file:

*C:\Program Files (x86)\BigFix Enterprise\BES Client__BESData__Global\Logs**YYYYMMDD.log***

# 6 Important Post-installation Steps

For security reasons and to ensure optimal operation of the deployment, it is recommended that you carefully read all the contents of this section, applying the indications and recommendations contained therein according to corporate needs and policies.

## 6.1 Secure the BigFix Site Admin Private Key File and Password

The site admin private key file "*C:\BESInstallers\license\license.pvk*" is present on the BigFix Server instance. The presence of this file is necessary for using the BigFix Administration Tool.

Be aware that anyone with access to the site admin private key file and its password can gain full control over the BigFix deployment, so ensure you keep both secured.

In addition, store a copy of the site admin private key file in a secure location, and make sure the password is not forgotten. This is critical because in case of loss of the site admin private key file, or in case the password is forgotten, there is no way to recover them, and it will therefore be necessary to reinstall the BigFix deployment from scratch using a new license.

## 6.2 Install the Client on the Microsoft SQL Server and on the Bastion Instances

It is recommended to install the Client on both the Microsoft SQL Server and the bastion instances so that they are also fully manageable through BigFix. This is not done automatically at stack creation time due to limitations concerning CloudFormation.

To install the Client, run the following steps on each instance:

1. Make sure the instance can resolve the hostname of the BigFix Server instance (as specified at CloudFormation launch time) to the corresponding private IP address (from the AWS

Console, the hostname can be found with key name "ServerName" on the "Parameters" tab of the stack, while the IP private address can be found with key name "BigFixServerPrivateIP" on the "Outputs" tab of the stack).

2. Create a "*bfclient_install*" folder on the instance.
3. Copy the following files from the BigFix Server instance to the "*bfclient_install*" folder:
   "*C:\BESInstallers\BigFix-BES-Client.exe*"
   "*C:\BESInstallers\masthead.afxm*"
4. Run "*BigFix-BES-Client.exe*" and follow the wizard.

Shortly after the installation completes, verify that a new computer corresponding to the instance appears on the Console.

## 6.3 Configure the Manual Key Exchange on the TLR

Since for security reasons the TLR is installed as an authenticating relay, the first registration of newly installed Clients that would point directly to the TLR can only happen via manual key exchange procedure. This procedure is not necessary for newly installed Clients that will perform their first registration through Leaf Node Relays installed on-prem and not configured as authenticating relays.

In short, the manual key exchange procedure can be configured and run as follows:

- Configure: a single password or a list of one-time passwords will have to be defined on the TLR.
- Run: a first, password-based registration will have to be performed on newly installed Clients that would point directly to the TLR. The password can either be entered manually by running a command line on the client, or passed to it at first start-up time using the `_BESClient_SecureRegistration` setting as part of a configuration file named "*clientsettings.cfg*" on Windows clients, and "*besclient.config*" on UNIX/Linux clients.

Follow the links included above to learn how to configure and run the manual key exchange procedure in a complete and comprehensive manner.

## 6.4 Enable Persistent Connection on the TLR

Since communication from the AWS network to the corporate LAN may not be possible, to allow the TLR to promptly notify a Leaf Node Relay of the presence of new content, we will take advantage of the persistent connection feature between relays. This feature requires adding specific BigFix settings on both the TLR and the Leaf Node Relay.

This section deals with TLR settings, while LNR settings will be part of section **7.1**.

From the Console, enable the persistent connection on the TLR using "BES Support" fixlet "Persistent Connection: Enable Relay" (ID 3665).

Optionally, from the Console, add the following setting to the TLR computer:

```
_BESRelay_PersistentConnection_KeepAliveSeconds = <number of seconds>
```

The above setting defines the interval with which the TLR sends heartbeats to keep persistent connections alive. The default value is 600 (10 minutes). Consider setting a lower value if there are firewalls between the Relays that could close persistent connections due to inactivity.

## 6.5 Consider Adding the TLR to your Public and Corporate DNS Servers

Clients and Relays that will have to connect directly to the TLR will have to be able to resolve the TLR hostname (as specified at CloudFormation launch time) to its public Elastic IP address (from the AWS

Console, the TLR hostname can be found with key name "RelayName" on the "Parameters" tab of the stack, while the public Elastic IP address can be found with key name "BigFixRelayPublicIP" on the "Outputs" tab of the stack).

For Internet Clients (remote workers) to be able to connect directly to the TLR, consider adding an A record for the TLR to your public DNS server and point the TLR hostname to its public Elastic IP address.

For on-prem Leaf Node Relays to be able to connect to the TLR, consider doing the same on your corporate internal DNS server.

# 7 Manage the BigFix Deployment

This section is intended to provide information that is specific to the BigFix on AWS Marketplace case. For everything else (e.g. configurations, backup, recovery, maintenance, troubleshooting, support, etc.) you can refer to the official BigFix documentation that can be found at https://help.hcltechsw.com/bigfix/11.0/platform/welcome/BigFix_Platform_welcome.html.

## 7.1 Install Leaf Node Relays (On-premises)

To serve Clients located on-prem at corporate facilities it is necessary to install one or more on-prem Leaf Node Relays that must connect to the TLR. To find out how many on-prem Leaf Node Relays to install depending on the number of Clients to serve, refer to the "BigFix Performance & Capacity Planning Resources" that can be found at https://bigfix-mark.github.io/.

Information regarding the system requirements for installing a Relay can be found at https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0104120.

### 7.1.1 Install a Leaf Node Relay on Windows (Intel)

1. Identify the on-prem Windows (Intel) system where to install the Leaf Node Relay and create there a "*bfclient_install*" folder.
2. Make sure the identified system can resolve the hostname of the TLR to its public Elastic IP address (from the AWS Console, the TLR hostname can be found with key name "RelayName" on the "Parameters" tab of the stack, while the public Elastic IP address can be found with key name "BigFixRelayPublicIP" on the "Outputs" tab of the stack).
3. Copy file "*C:\Program Files (x86)\BigFix Enterprise\BES Server\ActionSite.afxm*" from the BigFix Server instance to the "*bfclient_install*" folder.
4. Download the installer of the Client for Windows (Intel) from https://support.bigfix.com/bes/release/11.0/patch2/, and save it to the "*bfclient_install*" folder.
5. Create a "*clientsettings.cfg*" text file as follows, and place it in the "*bfclient_install*" folder along with the other two files:

```
__RelaySelect_Automatic=0
__RelayServer1=https://<TLR hostname>:52311/bfmirror/downloads/
__RelayServer2=https://<TLR hostname>:52311/bfmirror/downloads/
_BESClient_PersistentConnection_Enabled=1
_BESRelay_PersistentConnection_OpenParent=1
_BESClient_SecureRegistration=<authenticating relay password>
```

6. Run the Client installer and follow the wizard.
7. Shortly after installing, verify that a new computer corresponding to the Windows system appears on the Console.

8. Install the Relay on the same system using "BES Support" fixlet "Install BigFix Relay (Version 11.0.2)" (ID 5635).
9. On the Leaf Node Relay computer, manually restart the Client service.
10. From the Console, disable the client persistent connection on the Leaf Node Relay computer using "BES Support" fixlet "Persistent Connection: Disable Client" (ID 3660).

### 7.1.2 Install a Leaf Node Relay on Linux

1. Identify the on-prem Linux system where to install the Leaf Node Relay.
2. Make sure the identified system can resolve the hostname of the TLR to its public Elastic IP address (from the AWS Console, the TLR hostname can be found with key name "RelayName" on the "Parameters" tab of the stack, while the public Elastic IP address can be found with key name "BigFixRelayPublicIP" on the "Outputs" tab of the stack).
3. Copy file "*C:\Program Files (x86)\BigFix Enterprise\BES Server\ActionSite.afxm*" from the BigFix Server instance to the identified on-prem system. Rename it as "*actionsite.afxm*" (all lowercase) and place it in the "*/etc/opt/BESClient/*" folder (create the folder if it doesn't exist). Verify that the file is owned by "root".
4. Download the Client for the identified Linux system from https://support.bigfix.com/bes/release/11.0/patch2/, and install it (more information can be found at Installing the Client on Linux). Do not start the Client up for now.
5. Copy file "*/var/opt/BESClient/besclient.config.default*" as "*/var/opt/BESClient/besclient.config*" and then edit it to add the following lines:

```
[Software\BigFix\EnterpriseClient\Settings\Client\__RelayServer1]
value                          = http://<TLR hostname>:52311/bfmirror/downloads/

[Software\BigFix\EnterpriseClient\Settings\Client\__RelayServer2]
value                          = http://<TLR hostname>:52311/bfmirror/downloads/

[Software\BigFix\EnterpriseClient\Settings\Client\__RelaySelect_Automatic]
value                          = 0

[Software\BigFix\EnterpriseClient\Settings\Client\_BESClient_PersistentConnection_Enabled]
value                          = 1

[Software\BigFix\EnterpriseClient\Settings\Client\_BESRelay_PersistentConnection_OpenParent]
value                          = 1

[Software\BigFix\EnterpriseClient\Settings\Client\_BESClient_SecureRegistration]
value                          = <authenticating relay password>
```

6. Start the Client up (`systemctl start besclient`). Shortly after starting, verify that a new computer corresponding to the Linux system appears on the Console.
7. Install the Relay on the same system using "BES Support" fixlet "Install BigFix Relay on Linux and UNIX (Version 11.0.2)" (ID 5636).
8. On the Leaf Node Relay computer, manually restart the Client process (`systemctl restart besclient`).
9. From the Console, disable the client persistent connection on the Leaf Node Relay computer using "BES Support" fixlet "Persistent Connection: Disable Client" (ID 3660).

## 7.2 Install Clients

In this section we will deal separately with the case of on-prem Clients (section **7.2.1**) and that of Internet Clients (section **7.2.2**) because in the two cases the Relay to connect to is different, and the BigFix settings that must be specified during the installation phase are different.

### 7.2.1 On-Premises

We will assume the Leaf Node Relay is not an authenticating relay. If it is, it will be necessary to also include a valid authenticating relay password in the client configuration file ("*clientsettings.cfg*" on Windows or "*besclient.config*" on Linux).

#### *7.2.1.1 Install a Client on Windows (Intel or ARM)*

1. Create a "*bfclient_install*" folder on the computer where you want to install the Client.
2. Make sure the computer can resolve the hostname of the Leaf Node Relay.
3. Copy file "*C:\Program Files (x86)\BigFix Enterprise\BES Server\ActionSite.afxm*" from the BigFix Server instance to the "*bfclient_install*" folder.
4. Download the installer of the Client for the relevant Windows platform from https://support.bigfix.com/bes/release/ and save it to the "bfclient_install" folder.
5. Create a "*clientsettings.cfg*" text file as follows, and place it in the "*bfclient_install*" folder along with the other two files:

```
__RelaySelect_Automatic=0
__RelayServer1=https://<LNR hostname>:52311/bfmirror/downloads/
__RelayServer2=https://<LNR hostname>:52311/bfmirror/downloads/
```

6. Run the installer and follow the wizard.
7. Shortly after installing, verify that a new computer corresponding to the Windows system appears on the Console.

#### *7.2.1.2 Install a Client on Linux*

1. Make sure the computer where you want to install the Client can resolve the hostname of the Leaf Node Relay.
2. Copy file "*C:\Program Files (x86)\BigFix Enterprise\BES Server\ActionSite.afxm*" from the BigFix Server instance to the Linux computer. Rename it as "*actionsite.afxm*" (all lowercase) and place it in the "*/etc/opt/BESClient/*" folder (create the folder if it doesn't exist). Verify that the file is owned by "root".
3. Download the installer of the Client for the relevant Linux platform from https://support.bigfix.com/bes/release/, and install it (more information can be found at Installing the Client on Linux). Do not start the Client up for now.
4. Copy file "*/var/opt/BESClient/besclient.config.default*" as "*/var/opt/BESClient/besclient.config*" and then edit it to add the following lines:

```
[Software\BigFix\EnterpriseClient\Settings\Client\__RelayServer1]
value                          = http://<LNR hostname>:52311/bfmirror/downloads/

[Software\BigFix\EnterpriseClient\Settings\Client\__RelayServer2]
value                          = http://<LNR hostname>:52311/bfmirror/downloads/

[Software\BigFix\EnterpriseClient\Settings\Client\__RelaySelect_Automatic]
value                          = 0
```

5. Start the Client up (`systemctl start besclient`). Shortly after starting, verify that a new computer corresponding to the Linux system appears on the Console.

#### *7.2.1.3 Install a Client on Mac*

1. Create a "*bfclient_install*" folder on the computer where you want to install the Client.
2. Make sure the computer can resolve the hostname of the Leaf Node Relay.

3. Copy file "*C:\Program Files (x86)\BigFix Enterprise\BES Server\ActionSite.afxm*" from the BigFix Server instance to the to the "*bfclient_install*" folder. Rename it as "*actionsite.afxm*" (all lowercase).
4. Download the Client for Mac from https://support.bigfix.com/bes/release/ and save it to the "bfclient_install" folder.
5. Create a "*clientsettings.cfg*" text file as follows, and place it in the "*bfclient_install*" folder along with the other two files:

```
__RelaySelect_Automatic=0
__RelayServer1=https://<LNR hostname>:52311/bfmirror/downloads/
__RelayServer2=https://<LNR hostname>:52311/bfmirror/downloads/
```

6. Install the Client (more information can be found at Installing the Client on Mac).
7. Shortly after installing, verify that a new computer corresponding to the Mac system appears on the Console.

### 7.2.2 Across the Internet

On Internet Clients that will connect directly to the TLR we will enable command polling. As a matter of fact, since the TLR is in AWS, it will not be able to send UDP notifications to them to notify for the presence of new content. Command polling tells the Client to check in for new content on a regular interval, recommended here to be 14400 seconds (4 hours). The shorter this interval is, the more responsive the environment will be, but with an increase of network traffic. This value can be adjusted according to corporate policy requirements.

#### *7.2.2.1 Install a Client on Windows (Intel or ARM)*

1. Create a "*bfclient_install*" folder on the computer where you want to install the Client.
2. Make sure the computer can resolve the hostname of the TLR.
3. Copy file "*C:\Program Files (x86)\BigFix Enterprise\BES Server\ActionSite.afxm*" from the BigFix Server instance to the "*bfclient_install*" folder.
4. Download the installer of the Client for the relevant Windows platform from https://support.bigfix.com/bes/release/ and save it to the "bfclient_install" folder.
5. Create a "*clientsettings.cfg*" text file as follows, and place it in the "*bfclient_install*" folder along with the other two files:

```
__RelaySelect_Automatic=0
__RelayServer1=https://<TLR hostname>:52311/bfmirror/downloads/
__RelayServer2=https://<TLR hostname>:52311/bfmirror/downloads/
_BESClient_SecureRegistration=<authenticating relay password>
_BESClient_Comm_CommandPollEnable=1
_BESClient_Comm_CommandPollIntervalSeconds=14400
```

6. Run the installer and follow the wizard.
7. Shortly after installing, verify that a new computer corresponding to the Windows system appears on the Console.

#### *7.2.2.2 Install a Client on Linux*

1. Make sure the computer where you want to install the Client can resolve the hostname of the TLR.
2. Copy file "*C:\Program Files (x86)\BigFix Enterprise\BES Server\ActionSite.afxm*" from the BigFix Server instance to the Linux computer. Rename it as "*actionsite.afxm*" (all lowercase)

and place it in the "*/etc/opt/BESClient/*" folder (create the folder if it doesn't exist). Verify that the file is owned by "root".

3. Download the installer of the Client for the relevant Linux platform from https://support.bigfix.com/bes/release/, and install it (more information can be found at Installing the Client on Linux). Do not start the Client up for now.
4. Copy file "*/var/opt/BESClient/besclient.config.default*" as "*/var/opt/BESClient/besclient.config*" and then edit it to add the following lines:

```
[Software\BigFix\EnterpriseClient\Settings\Client\__RelayServer1]
value                          = http://<TLR hostname>:52311/bfmirror/downloads/

[Software\BigFix\EnterpriseClient\Settings\Client\__RelayServer2]
value                          = http://<TLR hostname>:52311/bfmirror/downloads/

[Software\BigFix\EnterpriseClient\Settings\Client\__RelaySelect_Automatic]
value                          = 0

[Software\BigFix\EnterpriseClient\Settings\Client\_BESClient_SecureRegistration]
value                          = <authenticating relay password>

[Software\BigFix\EnterpriseClient\Settings\Client\_BESClient_Comm_CommandPollEnable]
value                          = 1

[Software\BigFix\EnterpriseClient\Settings\Client\_BESClient_Comm_CommandPollIntervalSeconds]
value                          = 1440
```

5. Start the Client up (`systemctl start besclient`). Shortly after starting, verify that a new computer corresponding to the Linux system appears on the Console.

#### *7.2.2.3 Install a Client on Mac*

1. Create a "*bfclient_install*" folder on the computer where you want to install the Client.
2. Make sure the computer can resolve the hostname of the TLR.
3. Copy file "*C:\Program Files (x86)\BigFix Enterprise\BES Server\ActionSite.afxm*" from the BigFix Server instance to the "*bfclient_install*" folder. Rename it as "*actionsite.afxm*" (all lowercase).
4. Download the Client for Mac from https://support.bigfix.com/bes/release/ and save it to the "bfclient_install" folder.
5. Create a "*clientsettings.cfg*" text file as follows, and place it in the "*bfclient_install*" folder along with the other two files:

```
__RelaySelect_Automatic=0
__RelayServer1=https://<TLR hostname>:52311/bfmirror/downloads/
__RelayServer2=https://<TLR hostname>:52311/bfmirror/downloads/
_BESClient_SecureRegistration=<authenticating relay password>
_BESClient_Comm_CommandPollEnable=1
_BESClient_Comm_CommandPollIntervalSeconds=14400
```

6. Install the Client (more information can be found at Installing the Client on Mac).
7. Shortly after installing, verify that a new computer corresponding to the Mac system appears on the Console.

## 7.3 Add Top-Level Relays on AWS

If the size of the BigFix deployment increases over time, you might need to install another TLR on AWS. If you need to do so, follow these steps:

1. Create a new AWS EC2 instance as follows:

a. Choose an OS supported by the Relay component (more information at https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0104120).
b. Choose instance type and disk size according to the "BigFix Performance & Capacity Planning Resources" document that can be found at https://bigfix-mark.github.io/ (refer specifically to the "Top Level Relays" section).
c. Create the instance on the same VPC and public subnet as the existing TLR.
d. Select the same security group as the existing TLR.

2. Once the new instance has been created, associate an Elastic IP address with it (NOTE: This is required because a Relay must have a static IP address).
3. Install the Client on the new instance (follow the steps described in section **6.2**).
4. Once the Client is up and running, add the Relay component using "BES Support" fixlet "Install BigFix Relay (Version 11.0.2)" (ID 5635) or "Install BigFix Relay on Linux and UNIX (Version 11.0.2)" (ID 5636) depending on whether the new instance is Windows-based or Linux-based.
5. Enable relay authentication on the new instance using "BES Support" fixlet "BES Client Setting: Enable Relay Authentication" (ID 1297).
6. Configure the Manual Key Exchange as described in section **6.3**.
7. Enable persistent connection as described in section **6.4**.
8. Consider adding the new TLR to your public and corporate DNS servers (ref. section **6.5**).

# 8 Upgrade of the BigFix Deployment

When the BigFix Team will release new versions of BigFix, it will be possible to upgrade the BigFix deployment following the same instructions, recommendations and procedures that apply to full on-prem deployments, and which are documented at https://help.hcltechsw.com/bigfix/11.0/platform/Platform/Installation/c_upgrading1.html.

In particular, it will be possible to perform an automatic upgrade of the server components (Server, Web Reports, WebUI and Console), as well as a manual one. In any case, before starting the upgrade, it is always recommended to run the "BigFix Pre Upgrade Check (Version x.x.x)" fixlet targeting the instance that runs the server components to verify the existence of the prerequisites for upgrading them to the desired version.

## 8.1 Automatic Upgrade of the Server Components

The automatic upgrade of the server components uses fixlet "BigFix - Updated Platform Server Components version x.x.x now available!" of "BES Support". When running the fixlet, choose the "Dynamic targeting by properties" option, and then select "All computers". This targeting option will upgrade the server components on the BigFix Server instance first, and then, within 6 hours at most and without further user intervention, the Console on the BigFix Console instance as well.

The upgrade of the Console can also be done manually by downloading the installer of the desired version from https://support.bigfix.com/bes/release directly on the BigFix Console instance, and then running it and following the installation wizard.

## 8.2 Manual Upgrade of the Server Components

More information about the manual upgrade of BigFix components can be found at https://help.hcltechsw.com/bigfix/11.0/platform/Platform/Installation/c_manual_upgrade.html.

NOTE: Specifically in relation to the manual upgrade of the server components, it must be specified that the BigFix Server instance doesn't contain an initial version of the "BigFix Installation

Generator", therefore during the execution of the procedure documented at Upgrading the installation generator and the primary server the users will be presented with a "Setup Type" dialog on which they will have to choose the option "I want to install with an existing masthead", and then select file "*C:\BESInstallers\masthead.afxm*".

# Appendix A – Architecture Diagram and Description of Stack Resources

The following figure depicts the architecture diagram of "HCL BigFix Remediate v11" on AWS Marketplace:

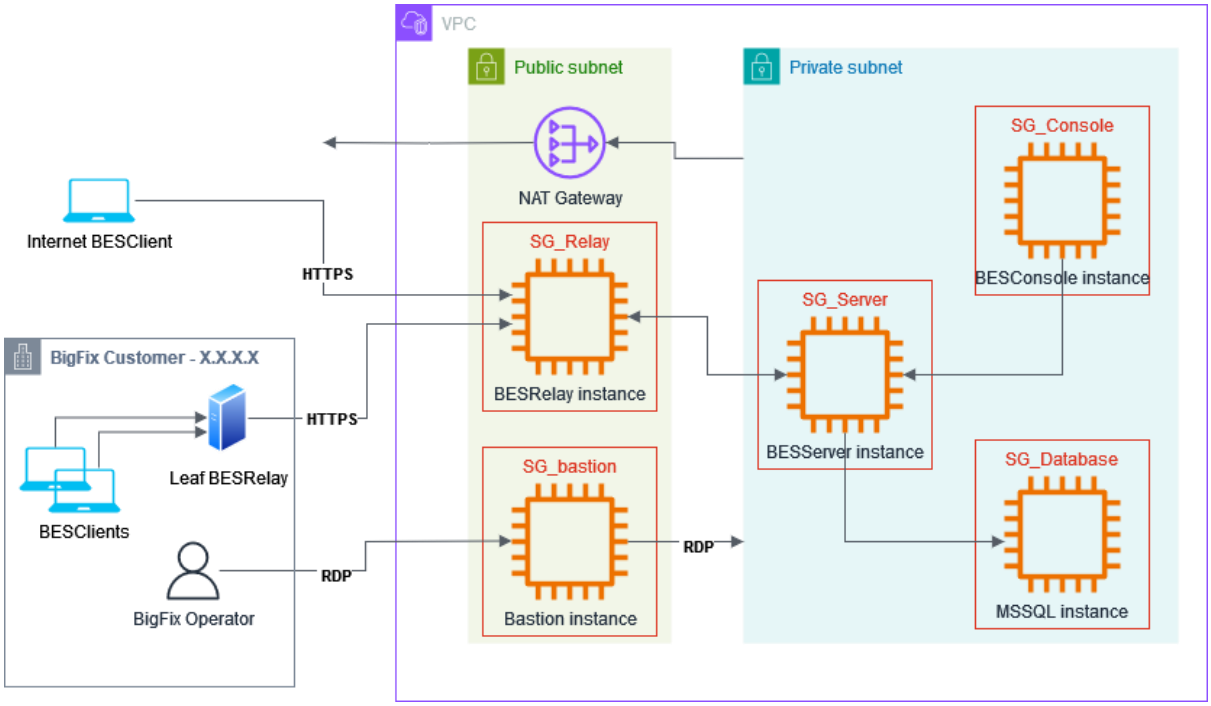


*Figure 2: Architecture diagram of "HCL BigFix Remediate v11" on AWS Marketplace.*

All stack resources are created in a single Availability Zone of the chosen AWS Region. The Availability Zone is selected automatically.

This is the list of stack resources – divided by service and resource type – created by CloudFormation (the number in brackets indicates how many resources of that type are created):

AWS::EC2::EIP *(2)*
AWS::EC2::Instance *(5)*
AWS::EC2::InternetGateway *(1)*
AWS::EC2::NatGateway *(1)*
AWS::EC2::Route *(2)*
AWS::EC2::RouteTable *(2)*
AWS::EC2::SecurityGroup *(5)*
AWS::EC2::Subnet *(2)*
AWS::EC2::SubnetRouteTableAssociation *(2)*
AWS::EC2::VPC *(1)*
AWS::EC2::VPCGatewayAttachment *(1)*
AWS::IAM::InstanceProfile *(1)*
AWS::IAM::Role *(1)*

The relationship between stack resources is represented in the following figure:


*Figure 3: Stack canvas exported from CloudFormation Application Composer.*

## A.1 – Description of the EC2 Instances

All EC2 instances are based on **Windows Server 2022 Datacenter** (64-bit).

### A.1.1 – Bastion Instance

- Doesn't run any BigFix components upon creation.
- Can be accessed through RDP from over the Internet using its public IPv4 address (whose value can be found from the AWS Console, by browsing the running instances via EC2 service).
- It is the instance that BigFix Administrators and Operators must connect to and use as bridge to access the other instances created by CloudFormation (ref. sections from **4.1.2** through **4.1.4**).
- Resides on the public subnet and has both a public and a private IPv4 address.

  NOTE: The public IPv4 address is dynamic, so it can change over time (this can typically happen upon system reboot). The current value can always be found from the AWS Console, by browsing the running instances via EC2 service.

- It's automatically created as a "t2.micro" type of instance, with a 30 GB disk of "gp2" type. These parameters can later be updated by the user.
- Its security group is named "<stack name>-bfbastion-sg". Go to appendix **A.2** for further details.

### A.1.2 – BigFix Server Instance

- Runs the following BigFix components: Server, Web Reports, WebUI and Client.
- Resides on the private subnet.
- Can be accessed through RDP from the bastion instance using the private IPv4 address (whose value can be found from the AWS Console, by browsing the running instances via EC2

service. Alternatively, the value is also available with key name "BigFixServerPrivateIP" on the "Outputs" tab of the stack).

- Doesn’t have a public IP address.
- Accesses the Internet through the NAT Gateway to gather BigFix license updates, and BigFix external sites content.
- Its security group is named "<stack name>-bfserver-sg". Go to appendix **A.2** for further details.

### A.1.3 – BigFix Console Instance

- Runs the following BigFix components: Console and Client.
- Resides on the private subnet.
- Can be accessed through RDP from the bastion instance using the private IPv4 address (whose value can be found from the AWS Console, by browsing the running instances via EC2 service. Alternatively, the value is also available with key name "BigFixConsolePrivateIP" on the "Outputs" tab of the stack).
- Doesn’t have a public IP address.
- Can access the Internet through the NAT Gateway.
- Its security group is named "<stack name>-bfconsole-sg". Go to appendix **A.2** for further details.

### A.1.4 – Microsoft SQL Server Instance

- Doesn’t run any BigFix components upon creation.
- Runs Microsoft SQL Server 2022 - Standard Edition (64-bit)
- Resides on the private subnet.
- Can be accessed through RDP from the bastion instance using the private IPv4 address (whose value can be found from the AWS Console, by browsing the running instances via EC2 service. Alternatively, the value is also available with key name "DatabaseInstancePrivateIP" on the "Outputs" tab of the stack).
- Doesn’t have a public IP address.
- Can access the Internet through the NAT Gateway.
- Its security group is named "<stack name>-bfdatabase-sg". Go to appendix **A.2** for further details.

### A.1.5 – BigFix Relay Instance

- Runs the following BigFix components: Relay and Client. The Relay is an authenticating one.
- Resides on the public subnet and has both a public and a private IPv4 address (whose values can be found from the AWS Console, by browsing the running instances via EC2 service. Alternatively, both values are also available on the "Outputs" tab of the stack, the former with key name "BigFixRelayPublicIP" and the latter with key name "BigFixRelayPrivateIP").

  NOTE: The public IPv4 address is an "Elastic IP", so it is static. This is critical because that’s the IP address that Internet Clients (remote workers) and on-prem Leaf Node Relays must always use to connect to the Relay, so it must not change over time.

- Can be accessed through RDP from the bastion instance using the private IPv4 address.
- Its security group is named "<stack name>-bfrelay-sg". Go to appendix **A.2** for further details.

## A.2 – Description of the Security Groups

Note that in place of the following 3 generic expressions that are used in the "Source" column of the below tables, the AWS Console will show the corresponding values specified by the user at CloudFormation launch time (ref. **3.2**):

- Private subnet CIDR block
- Public subnet CIDR block
- Inbound source for RDP access

### A.2.1 – Bastion Instance Security Group

From the AWS Console, the security group of the bastion instance can be found with key name "<stack name>-bfbastion-sg" on the "Resources" tab of the stack. In the architecture diagram of *Figure 2*, it is indicated as "SG_bastion".

| Bastion instance – Inbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Source | Description |
| Custom UDP | UDP | 52311 | Private subnet CIDR block | Allows a possible local BigFix Client to be notified by the BigFix Server. |
| RDP | TCP | 3389 | Inbound source for RDP access | Allows RDP access from the source specified by the user at stack creation time. |

*Figure 4: Bastion instance – Security group – Inbound rules.*

| Bastion instance – Outbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Destination | Description |
| HTTPS | TCP | 443 | 0.0.0.0/0 | Allows outbound traffic to the Internet limited to TCP/443. |
| All traffic | All | All | Private subnet CIDR block | Allows outbound traffic to the private subnet. |
| All traffic | All | All | Public subnet CIDR block | Allows outbound traffic to the public subnet. |

*Figure 5: Bastion instance – Security group – Outbound rules.*

### A.2.2 – BigFix Server Instance Security Group

From the AWS Console, the security group of the BigFix Server instance can be found with key name "<stack name>-bfserver-sg" on the "Resources" tab of the stack. In the architecture diagram of *Figure 2*, it is indicated as "SG_Server".

| BigFix Server instance – Inbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Source | Description |
| Custom TCP | TCP | 52311 | Private subnet CIDR block | Allows BigFix Clients on the private subnet to connect to the BigFix Server. |
| Custom TCP | TCP | 52311 | Public subnet CIDR block | Allows the BigFix Relay (TLR) to connect to the BigFix Server. |
| Custom TCP | TCP | 8083 | Private subnet CIDR block | Allows access to BigFix Web Reports from the private subnet. |
| HTTPS | TCP | 443 | Private subnet CIDR block | Allows access to BigFix WebUI from the private subnet. |
| RDP | TCP | 3389 | Public subnet CIDR block | Allows RDP access from the bastion instance. |

*Figure 6: BigFix Server instance – Security group – Inbound rules.*

| BigFix Server instance – Outbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Destination | Description |
| Custom UDP | UDP | 52311 | Public subnet CIDR block | Allows the BigFix Server to notify possible BigFix Clients on the public subnet. |
| Custom TCP | TCP | 52311 | Public subnet CIDR block | Allows the BigFix Server to notify the BigFix Relay (TLR). |
| HTTPS | TCP | 443 | 0.0.0.0/0 | Allows outbound traffic to the Internet limited to TCP/443. Required to gather BigFix license updates and external sites content. |
| HTTP | TCP | 80 | 0.0.0.0/0 | Allows outbound traffic to the Internet limited to TCP/80. Required to prefetch files with HTTP URLs in the Action Script. |
| All traffic | All | All | Private subnet CIDR block | Allows outbound traffic to the private subnet. |

*Figure 7: BigFix Server instance – Security group – Outbound rules.*

## A.2.3 – BigFix Console Instance Security Group

From the AWS Console, the security group of the BigFix Console instance can be found with key name "<stack name>-bfconsole-sg" on the "Resources" tab of the stack. In the architecture diagram of *Figure 2*, it is indicated as "SG_Console".

| BigFix Console instance – Inbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Source | Description |
| Custom UDP | UDP | 52311 | Private subnet CIDR block | Allows the BigFix Client to be notified by the BigFix Server. |
| RDP | TCP | 3389 | Public subnet CIDR block | Allows RDP access from the bastion instance. |

*Figure 8: BigFix Console instance – Security group – Inbound rules.*

| BigFix Console instance – Outbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Destination | Description |
| Custom TCP | TCP | 52311 | Public subnet CIDR block | Allows access to the "Relay Diagnostics" page of the BigFix Relay (TLR). |
| HTTPS | TCP | 443 | 0.0.0.0/0 | Allows outbound traffic to the Internet limited to TCP/443. |
| All traffic | All | All | Private subnet CIDR block | Allows outbound traffic to the private subnet. |

*Figure 9: BigFix Console instance – Security group – Outbound rules.*

## A.2.4 – Microsoft SQL Server Instance Security Group

From the AWS Console, the security group of the Microsoft SQL Server instance can be found with key name "<stack name>-bfdatabase-sg" on the "Resources" tab of the stack. In the architecture diagram of *Figure 2*, it is indicated as "SG_Database".

| Microsoft SQL Server instance – Inbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Source | Description |
| Custom UDP | UDP | 52311 | Private subnet CIDR block | Allows a possible local BigFix Client to be notified by the BigFix Server. |
| MSSQL | TCP | 1433 | Private subnet CIDR block | Allows BigFix Server, Web Reports and WebUI to access the database. |
| RDP | TCP | 3389 | Public subnet CIDR block | Allows RDP access from the bastion instance. |

*Figure 10: Microsoft SQL Server instance – Security group – Inbound rules.*

| Microsoft SQL Server instance – Outbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Destination | Description |
| HTTPS | TCP | 443 | 0.0.0.0/0 | Allows outbound traffic to the Internet limited to TCP/443. |
| All traffic | All | All | Private subnet CIDR block | Allows outbound traffic to the private subnet. |

*Figure 11: Microsoft SQL Server instance – Security group – Outbound rules.*

## A.2.5 – BigFix Relay Instance Security Group

From the AWS Console, the security group of the BigFix Relay instance can be found with key name "<stack name>-bfrelay-sg" on the "Resources" tab of the stack. In the architecture diagram of *Figure 2*, it is indicated as "SG_Relay".

| BigFix Relay instance – Inbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Source | Description |
| Custom TCP | TCP | 52311 | 0.0.0.0/0 | Allows BigFix communications from the Internet (i.e. BigFix Clients and Relays installed by the customer), and from the BigFix Server. |
| Custom UDP | UDP | 52311 | Private subnet CIDR block | Allows the BigFix Server to notify the local BigFix Client in case the BigFix Relay is stopped. |
| RDP | TCP | 3389 | Public subnet CIDR block | Allows RDP access from the bastion instance. |

*Figure 12: BigFix Relay instance – Security group – Inbound rules.*

| BigFix Relay instance – Outbound Rules | | | | |
|---|---|---|---|---|
| Type | Protocol | Port range | Destination | Description |
| HTTPS | TCP | 443 | 0.0.0.0/0 | Allows outbound traffic to the Internet limited to TCP/443. |
| All traffic | All | All | Private subnet CIDR block | Allows outbound traffic to the private subnet. |

*Figure 13: BigFix Relay instance – Security group – Outbound rules.*

## A.3 – Description of the IAM Role and IAM Instance Profile

To allow the BigFix Server instance to download the BigFix license authorization file ("**.BESLicenseAuthorization*") from an S3 bucket, CloudFormation will create the following two IAM resources:

- A **IAM role** named "<stack-name>-role", to which the AWS-managed permission policy "AmazonS3ReadOnlyAccess" is then attached. The trust policy of the role allows EC2 instances to assume it.
- A **IAM instance profile** named "<stack-name>-profile" to which the above IAM role is added. The IAM instance profile is then associated with the BigFix Server instance, eventually allowing it to assume the IAM role.

NOTE: Although not done automatically by CloudFormation due to limitations inherent to the service, once the stack creation has been successfully completed it is possible to delete the IAM role "<stack-name>-role" manually. Note that deleting the IAM role will also automatically delete the IAM instance profile "<stack-name>-profile".

## A.4 – Description of the Elastic IP Addresses

CloudFormation will create two Elastic IP addresses, and then associate one with the BigFix Relay instance, and the other one with the NAT gateway.

## A.5 – Description of Other Network Resources

In addition to the stack resources described in the previous sections, CloudFormation will also create the following ones, which all together form the network infrastructure on which the core BigFix deployment hosted on AWS relies:

- **1 VPC** (Virtual Private Cloud). It will use the IPv4 CIDR block specified by the user at CloudFormation launch time.
- **2 subnets**, one private and one public. Each one will use the corresponding IPv4 subnet CIDR block specified by the user at CloudFormation launch time.
    - Instances in the private subnet can connect to the Internet but cannot receive unsolicited inbound connections from the Internet.
    - Instances in the public subnet can connect to the Internet and can also receive inbound connections from the Internet if the connections satisfy the rules of the security group associated with the instance.
- **1 Internet gateway**. It is attached to the VPC. It allows communication between resources in the public subnet and the Internet. This communication is achieved by creating a route table with an appropriately defined route that targets the Internet gateway, and then associating the table to the public subnet. All these operations are performed by CloudFormation.
- **1 NAT gateway**. It is created in the public subnet. It allows resources in the private subnet to connect to the Internet (outbound only). This connection is achieved by creating a route table with an appropriately defined route that targets the NAT gateway, and then associating the table to the private subnet. All these operations are performed by CloudFormation.

# Appendix B – IAM Permissions

This appendix section describes the IAM permissions that CloudFormation requires to be able to perform all stack operations.

These permissions can be granted to CloudFormation through a customer-managed IAM policy attached to a IAM service role and then letting CloudFormation assume that role at stack creation time (ref. **Step 3 – "Configure stack options"**).

NOTE: When creating the IAM service role, make sure you specify a trust policy that would allow it to be assumed by CloudFormation.

Next appendix sections **B.1** and **B.2** contain permission sets that – when put all together – should represent a good reference as to what CloudFormation requires to be able to perform all stack operations.

## B.1 – IAM Permissions for Marketplace, CloudFormation, EC2 and S3 Services

The following permission set for the **Marketplace**, **CloudFormation**, **EC2** and **S3** services represents a valid reference as it allowed (along with the permission sets of appendix section **B.2**) to successfully complete all test phases (NOTE: Replace "xx-xxxxxx-x" with the chosen AWS Region):

```
"Effect": "Allow",
"Action": [
    "aws-marketplace:Describe*",
    "aws-marketplace:List*",
    "aws-marketplace:View*",
    "cloudformation:*",
    "ec2:AllocateAddress",
    "ec2:AssociateAddress",
    "ec2:AssociateIamInstanceProfile",
    "ec2:AssociateRouteTable",
    "ec2:AttachInternetGateway",
    "ec2:AuthorizeSecurityGroup*",
    "ec2:CreateDefaultSubnet",
    "ec2:CreateEgressOnlyInternetGateway",
    "ec2:CreateInternetGateway",
    "ec2:CreateKeyPair",
    "ec2:CreateNatGateway",
    "ec2:CreateRoute",
    "ec2:CreateRouteTable",
    "ec2:CreateSecurityGroup",
    "ec2:CreateSubnet*",
    "ec2:CreateVpc*",
    "ec2:DeleteEgressOnlyInternetGateway",
    "ec2:DeleteInternetGateway",
    "ec2:DeleteNatGateway",
    "ec2:DeleteRoute",
    "ec2:DeleteRouteTable",
    "ec2:DeleteSecurityGroup",
    "ec2:DeleteSubnet*",
    "ec2:DeleteVpc*",
    "ec2:Describe*",
    "ec2:DetachInternetGateway",
    "ec2:DisassociateAddress",
    "ec2:DisassociateRouteTable",
    "ec2:ModifySecurityGroupRules",
    "ec2:ModifySubnetAttribute",
    "ec2:ModifyVpcAttribute",
    "ec2:ReleaseAddress",
    "ec2:ReplaceIamInstanceProfileAssociation",
    "ec2:ReplaceRoute",
    "ec2:ReplaceRouteTableAssociation",
    "ec2:RevokeSecurityGroup*",
    "ec2:RunInstances",
    "ec2:*Tags*",
    "ec2:TerminateInstances",
    "ec2:UpdateSecurityGroupRuleDescriptions*",
    "s3:*"
],
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "xx-xxxxxx-x"
    }
}
```

## B.2 – IAM Permissions for the IAM Service

The following 3 permission sets for the **IAM** service represent a valid reference as they allowed (all together and along with the permissions of appendix section **B.1**) to successfully complete all test phases:

Set 1:

```
"Effect": "Allow",
"Action": [
    "iam:CreateRole",
    "iam:DeleteRole",
    "iam:Get*",
    "iam:*InstanceProfile*",
    "iam:List*",
    "iam:PassRole"
],
"Resource": "*"
```

Set 2:

```
"Effect": "Allow",
"Action": [
    "iam:AttachRolePolicy",
    "iam:DetachRolePolicy"
],
"Resource": "*",
"Condition": {
    "ArnEquals": {
        "iam:PolicyARN": "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess"
    }
}
```

Set 3:

```
"Effect": "Allow",
"Action": "iam:CreateServiceLinkedRole",
"Resource": "*",
"Condition": {
    "StringLike": {
        "iam:AWSServiceName": [
"ec2.amazonaws.com",
"s3.amazonaws.com"
        ]
    }
}
```

# Appendix C – Recommended Instance Type and Disk Size by BigFix Instance

General information provided by AWS can be found at the following links:

**Amazon EC2 instance types**: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html?icmpid=docs_ec2_console

**Amazon EC2 On-Demand Pricing**: https://aws.amazon.com/ec2/pricing/on-demand/

**AWS Pricing Calculator**: https://calculator.aws/#/addService

## C.1 – Recommended Values for the BigFix Server Instance

In the following table, the column "# of Clients" refers to the total number of Clients installed across the whole BigFix deployment:

| # of Clients | Instance Type | Disk Size (GB) |
|---|---|---|
| up to 1,000 | **t3.xlarge** (4 CPU / 16 GB RAM) | 100 |
| up to 5,000 | **t3.xlarge** (4 CPU / 16 GB RAM) | 120 |
| up to 10,000 | **c5.2xlarge** (8 CPU / 16 GB RAM) | 150 |
| up to 50,000 | **c5.2xlarge** (8 CPU / 16 GB RAM) | 200 |
| up to 200,000 | **m5a.4xlarge** (16 CPU / 64 GB RAM) | 450 |

*Figure 14: BigFix Server instance – Recommended instance type and disk size.*

## C.2 – Recommended Values for the BigFix Relay Instance

In the following table, the column "# of Clients" refers to the total number of Clients managed by the Relay:

| # of Clients | Instance Type | Disk Size (GB) |
|---|---|---|
| up to 1,000 | **t3.medium** (2 CPU / 4 GB RAM) | 100 |
| up to 5,000 | **t3.medium** (2 CPU / 4 GB RAM) | 120 |
| up to 10,000 | **t3.medium** (2 CPU / 4 GB RAM) | 150 |
| up to 40,000 | **c5.xlarge** (4 CPU / 8 GB RAM) | 150 |
| more than 40,000 | Add more Top-Level Relays (ref. section 7.3) | N/A |

*Figure 15: BigFix Relay instance – Recommended instance type and disk size.*

## C.3 – Recommended Values for the BigFix Console Instance

In the following table, the column "# of Clients" refers to the total number of Clients installed across the whole BigFix deployment:

| # of Clients | Instance Type | Disk Size (GB) |
|---|---|---|
| up to 1,000 | **t3.xlarge**<br>(4 CPU / 16 GB RAM) | 50 |
| up to 5,000 | **t3.xlarge**<br>(4 CPU / 16 GB RAM) | 50 |
| up to 10,000 | **c5.2xlarge**<br>(8 CPU / 16 GB RAM) | 50 |
| up to 50,000 | **c5.4xlarge**<br>(16 CPU / 32 GB RAM) | 80 |
| up to 200,000 | **m5zn.6xlarge**<br>(24 CPU / 96 GB RAM) | 240 |

*Figure 16: BigFix Console instance – Recommended instance type and disk size.*

## C.4 – Recommended Values for the Microsoft SQL Server Instance

In the following table, the column "# of Clients" refers to the total number of Clients installed across the whole BigFix deployment:

| # of Clients | Instance Type | Disk Size (GB) |
|---|---|---|
| up to 1,000 | **m5.xlarge**<br>(4 CPU / 16 GB RAM) | 120 |
| up to 5,000 | **m5.xlarge**<br>(4 CPU / 16 GB RAM) | 150 |
| up to 10,000 | **c5.2xlarge**<br>(8 CPU / 16 GB RAM) | 250 |
| up to 50,000 | **m5.2xlarge**<br>(8 CPU / 32 GB RAM) | 300 |
| up to 200,000 | **m5a.4xlarge**<br>(16 CPU / 64 GB RAM) | 1000 |

*Figure 17: Microsoft SQL Server instance – Recommended instance type and disk size.*