

**BigFix**  
**BigFix 10 Insights Getting Started Guide**



# Special notice

Before using this information and the product it supports, read the information in Notices.

# Edition notice

This edition applies to BigFix version 10 and to all subsequent releases and modifications until otherwise indicated in new editions.

# Contents

<b>Chapter 1. Version History.....</b>	<b>1</b>
<b>Chapter 2. Introduction.....</b>	<b>2</b>
<b>Chapter 3. Prerequisites and limitations.....</b>	<b>5</b>
<b>Chapter 4. Setting up BigFix Insights.....</b>	<b>37</b>

# Chapter 1. Version History

<b>Date</b>	<b>Version</b>	<b>Note</b>
30/09/2020	1.0	Initial release
30/11/2021	2.0	Document split into Getting Started and Operating Guides

# Chapter 2. Introduction

BigFix Insights is an application-driven extract, transform, and load (ETL) that helps you consolidate all the BigFix data into a single data warehouse.

Insights creates an automatically correlated view of all your devices and BigFix data across all ETLed deployments. A Business Intelligence (BI) tool can then reference the resulting Insights database to generate analytics and visualizations needed to report on and mitigate risks.

Insights provides an ETL that is capable of warehousing and aggregating data from multiple BigFix deployments. From this data warehouse, you can utilize business intelligence and analytics tooling such as Tableau, and Power BI.

## Features and capabilities

Following is the list of features and capabilities that BigFix Insights offers:

- Import and consolidate data from BigFix Enterprise data sources into a single data lake with the data optimized for report generation.
- Provide a WebUI-based tool to manage ETL scheduling.
- Leverage a BI reporting tool to provide out-of-the-box reports showing data summaries grouped by various criteria and historical trending, with a rich set of visualization and dynamic filtering capabilities to cover the following areas:
  - Patch rhythm
  - Device inventory
  - OS migration
  - Deployment progress

## Technical Background

The BigFix Enterprise Server (BFEnterprise) Database contains information regarding your Fixlets, analyses, computers, sites, and properties. Data stored within the BFEnterprise database is stored in a denormalized format optimized to support the near real time platform capabilities. Insights ingests the information from the BFEnterprise Database (replica) and transforms this data into a standard normalized format. This format is

optimized to support querying in SQL, reporting and servicing visualizations of important data across your organization. This document also outlines the schema associated with the BigFix Insights application.

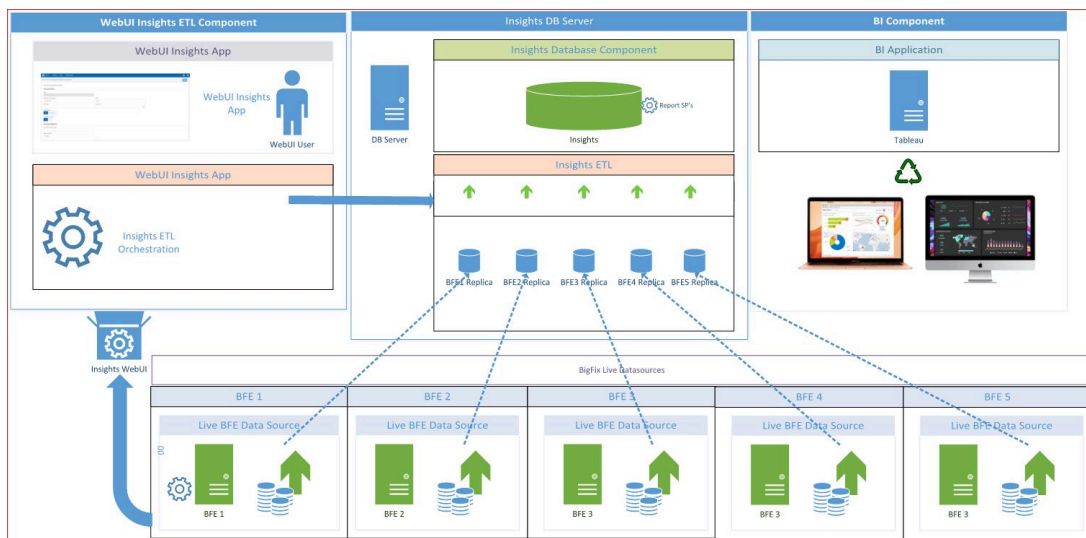
### Entitlement criteria

BigFix Insights is available as an add-on to any existing BigFix offering (Patch, Lifecycle, Start Kit, Compliance, or Inventory). For more details, contact your HCL account representative.

### Architecture overview

The below figure shows an architecture overview of BigFix Insights:

Figure 1. BigFix Insights - architecture overview



The above architecture illustrates the infrastructure components of BigFix Insights. The BigFix Insights application is built upon a MSSQL DBMS and it leverages the WebUI platform as the driving application. The Insights application ingests the structured data from one or many BigFix Enterprise Root Server Databases. To preserve the performance of the BigFix Root Server platform and mitigate potential impact on the application, the data is retrieved from a backup or replica (leveraging T-Log shipping) of the BFE Enterprise Database. Performing an ETL upon a live BFE Enterprise data source is not supported within the initial release and should be avoided in all production implementation scenarios. The Insights

Datalake schema is documented and provides a normalized format of the BFEnterprise Databases allowing customers to leverage a BI tool to develop reports.

Within the initial release, several reports are provided within tableau as an example of the capabilities. Tableau offers a wealth of data visualization options and facilitates a robust user interface for creating and developing reports, however customers can also leverage other BI interfaces.



# Chapter 3. Prerequisites and limitations

This section provides detailed information about the prerequisites and limitations for BigFix 10 Insights.

## System Prerequisites

The system requirements for BigFix 10 Insights are as follows:

- An entitlement to install and use BigFix Insights.
- BigFix Insights requires a Windows WebUI Server that is subscribed to the BigFix Insights content site.



### Notes:

- Insights consists of two BigFix content sites. The external content site for delivery of sample reports and the WebUI content site.
  - The WebUI content site must subscribe to the WebUI server that facilitates the ETL. The external content site must subscribe to a computer to retrieve the tableau sample workbooks. These sites collectively must not subscribe to all computers within the environment.
  - The Linux WebUI is currently not supported.
  - For WebUI configuration, please refer to [https://help.hcltechsw.com/bigfix/10.0/webui/WebUI/Admin\\_Guide/WebUI\\_admin\\_guide.html](https://help.hcltechsw.com/bigfix/10.0/webui/WebUI/Admin_Guide/WebUI_admin_guide.html)
- An SQL replica or offline copy of the BigFix Enterprise Database Ingestion data source.



### Notes:

- Do not run an Insights ETL against a live BigFix root server. A replica is required to serve as the datasource for BigFix Insights. The replica can be facilitated by a variety of SQL processes, including backups, T-Log shipping, and SQL *Always On*.



- Insights ETL supports ingesting only from Windows-based root servers (BigFix 10).
  - DB2 databases or root servers are not supported.
- BigFix Insights requires Microsoft SQL Server 2017 or DBMS server 2019 for the Insights Data Lake; prior versions are not supported.
  - Insights Database Server requires "sa" permissions.
  - Insights Data Lake must be generated from a Windows computer that is running WebUI.
  - To represent the Tableau reports that come with the product, a Tableau license agreement is required to run and view the provided Insights Tableau workbooks. This license is a separate entitlement from Insights. A minimum of one Tableau Creator license is required. You can use the explorer or view licenses to view workbooks.

## Credential prerequisites

The credential prerequisites for the Insights ETL components follow:

**Table 1. Credential prerequisites**

Required rights	Purpose
DB Reader to ingestion sources (REPLI-CAS)	The DB Reader is used to Authenticate and read information from data sources that are ingested by Insights.
DBO to Insights	The Insights account is use to maintain the application schema with an ongoing or persistent ETL and to create the Insights database.

## Network prerequisites

As with all data loading applications, minimize and optimize network latency and bandwidth impacts. Align resources within close network proximity, where or when possible. If possible, co-locate replicas with the Insights Data Lake.

**Table 2. Network rights**

Required rights	Purpose
Insights ETL Server requires clear line of sight to all ingested SQL Instances or Servers (replicas) over the listening SQL port that support BigFix Enterprise.	Insights Data Lake DB Server > Ingestion Source DB: <Ingestion Source DB Port>
The server that is running the WebUI instance requires connection to the Insights ETL Server Database by the 1433 port or the SQL listening port.	WebUI Server > Insights Data Lake DB Server:<Ingestion Source DB Port>

## Application requirements

### DBMS prerequisites

The DBMS prerequisites for Insights DBMS component follow:

**Table 3. DBMS prerequisites**

Required rights	Purpose
Minimum ingestion BigFix Enterprise version	<p>Insights ingests one or more BFE data sources.</p> <ul style="list-style-type: none"> <li>• The required data sources must be 9.5 or later.</li> <li>• Insights does not support ingestion of DB2 data sources.</li> </ul>
Insights ingestion to live BigFix Enterprise deployments is <b>not</b> supported.	<ul style="list-style-type: none"> <li>• In the initial release, the performance impact to BigFix core platform was reduced.</li> <li>• Insights must ingest data from an offline copy or replica of the BigFix Enterprise Database.</li> </ul>

**Table 3. DBMS prerequisites (continued)**

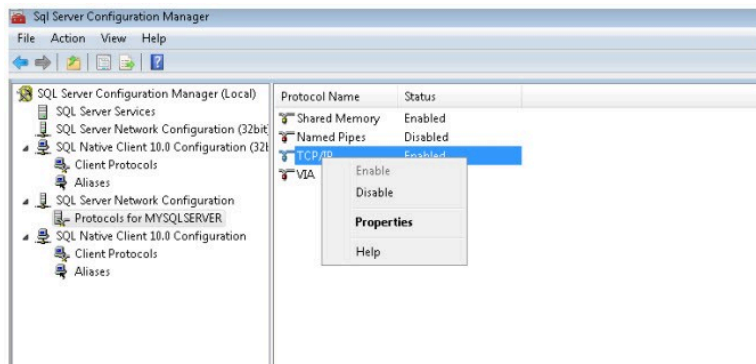
Required rights	Purpose
	Locate the offline copy of the database in close network proximity to the Insights database.

## SQL Server Listening Port

By default, SQL Server listens on TCP port number 1433, but for named instances the TCP port is dynamically configured. There are several options available to get the listening port for a SQL Server named instance.

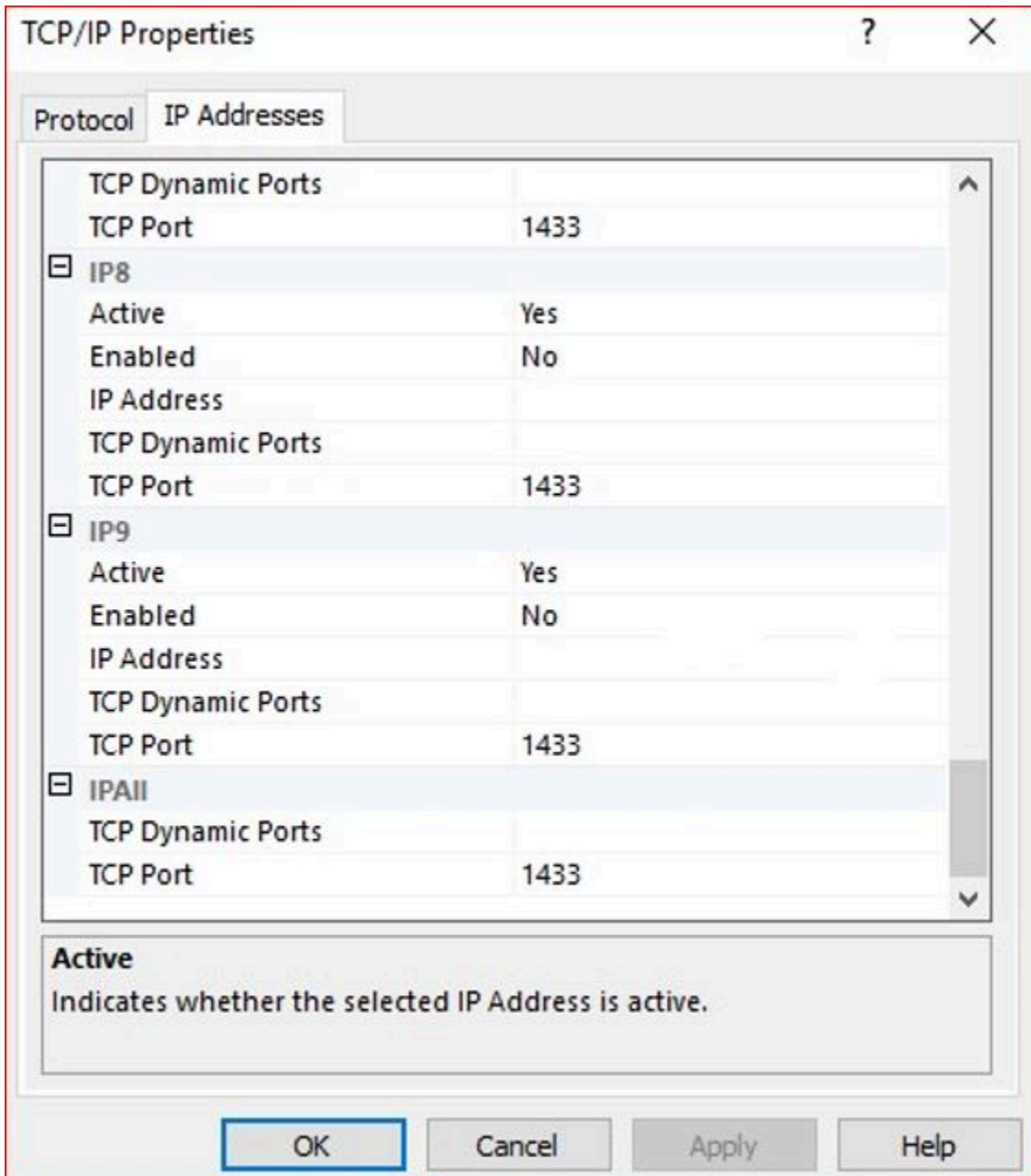
Execute the following steps to get the listening port information (SQL Server Configuration Manager):

1. Click **Start > All Programs > Microsoft SQL Server 2008 > Configuration Tools > SQL Server Configuration Manager**
2. Go to **SQL Server Configuration Manager > SQL Server Network Configuration > Protocols for <your Database>**
3. Right click **TCP/IP** and select **Properties**.



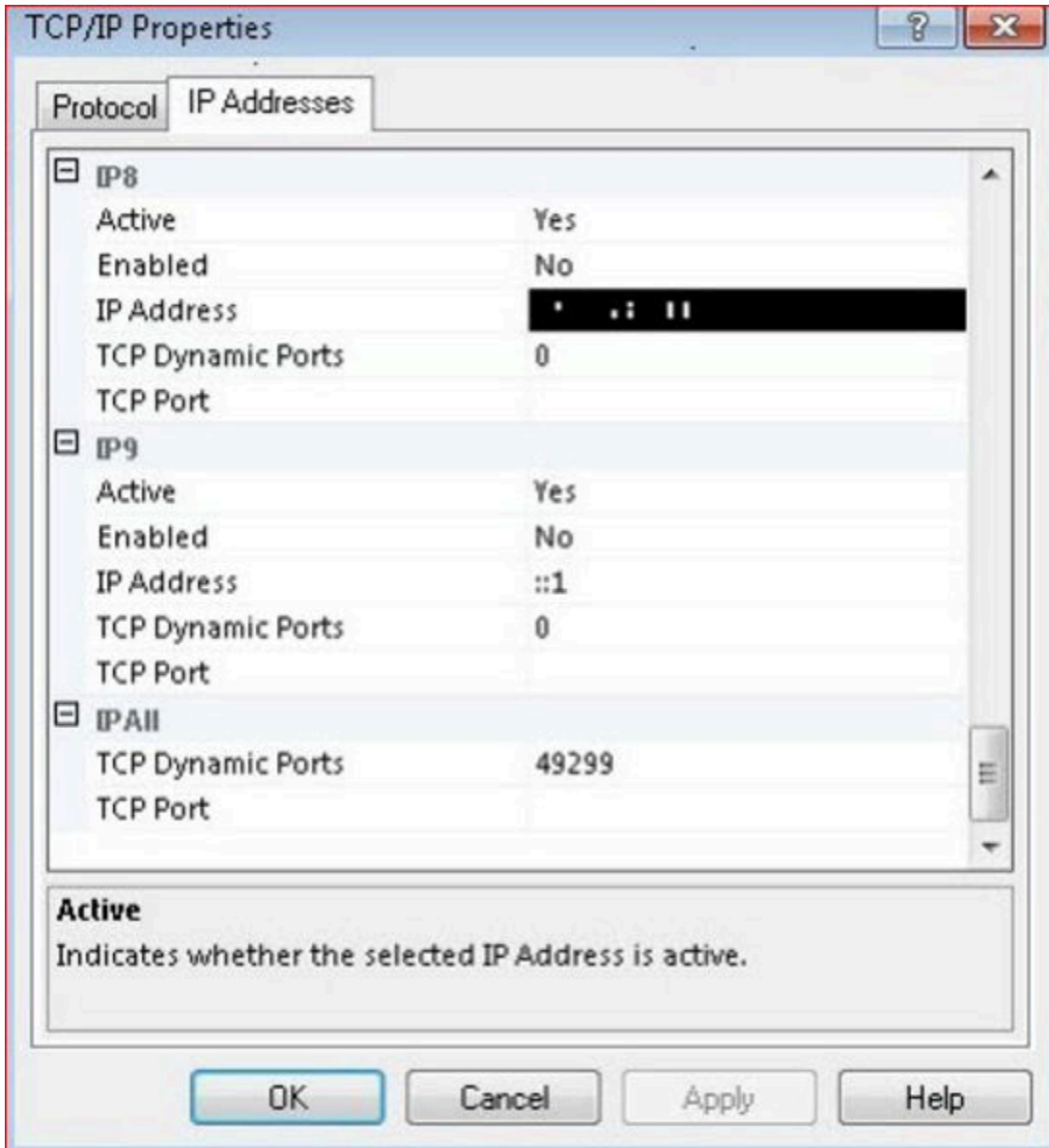
4. In the TCP/IP Properties dialog box, go to the IP Addresses tab and scroll down to the "IPAll" group.

If the configuration is on a static port, the current port number will be available in the TCP Port text box.



In the above screenshot, the instance is listening on port number 1433.

If SQL server is configured on a dynamic port, the current port number is available in the **TCP Dynamic Ports** text box.



In the above screenshot, the instance is listening on port number 49299.

## Compute and memory requirements

The following section is excerpted from the BigFix Capacity Planning guide. You can refer the complete publication: [BigFix Performance & Capacity Planning Resources](#)

For best capacity-planning results, base the Insights deployment on a database replica. You have many options for managing the base capacity, such as using database backups, MS SQL replication support, and MS SQL availability groups.

- For an MS SQL replica, additional MS SQL resources are required. See the following table for considerations and note the additional pressure on log space that is required for the replica.
- It is possible to bypass the replica for low-scale and test deployments. However, use care with this approach by using premonitoring and postmonitoring, in particular for the DBMS, to ensure system and database health.
- For input/output operations per second (IOPS), the general DBMS server standard of 5,000 IOPS with less than 1msec latency provides the best results. However, the I/O profile tends to be more read intensive and highly subject to reporting content and workloads. As a result, it might be possible to have good performance with a 2,500-IOPS based storage device. However, this possibility can only be assessed through careful monitoring.

For the Insights server, the capacity planning requirements are for the base BigFix offering.

- The requirements for the base operating system, the MS SQL secondary instance (if applicable), and the customer provided reporting technology (for example, Tableau), and associated data sources must be included.
- For best results, the deployment size is represented in the BigFix Enterprise computers table data cardinality for the initial offering.

The CPU and memory recommendations are based on the Extract-Transform-Load (ETL) agent that ingests the BigFix root server content to build the Insights Data Lake. These requirements are generally low because of the economy of the ETL process.

**Table 4. BigFix Insights MS SQL replica capacity planning requirements**

<b>Component</b>	<b>Additional CPUs</b>	<b>Additional memory in GB</b>	<b>Log space in GB</b>
------------------	------------------------	--------------------------------	------------------------

**Table 4. BigFix Insights MS SQL replica capacity planning requirements (continued)**

MS SQL Repl- cation	+2	+4	Two times more than the nor- mal space
------------------------	----	----	---

<b>Deployment size</b>	<b>CPU</b>	<b>Memory (GB)</b>	<b>Storage (GB)</b>
10,000	+4	+10	10% of BFEnterprise

## Workload management and Insights DBMS sizing

### Ingestion objects

To understand scalability implications in BigFix Insights, you must understand the objects that are imported from BigFix Enterprise Data sources. In this release, the following objects are imported:

- Actions
- Sites
- Computers
- Properties
- Property results
- Fixlets (analysis, tasks and fixlets), referred to as "content" in BigFix Insights
- Fixlet results, referred to as "content results" in BigFix Insights
- Groups (Including group membership)

In this release, the following root objects are not included in when you import Insights:

- Operators
- Roles

### Global content and linked items

In this release, global object mapping is limited to external content sites, fixlets (and tasks) and analysis. Custom content linking is not within the scope of this release.



In the Insights database, you can import data from one or more BigFix Enterprise Database data sources. This release supports data ingestion from up to 10 data sources, and the total number of endpoints of the ingested data sources must not exceed 250,000.

The concept of global objects amounts to the singular representation of the same object from multiple BigFix Enterprise Database data sources. For example, the BigFix Enterprise Server support site is the same on one data source, many data sources, and every data source.

## Sizing concepts

The scalability and size requirements of Insights is directly related to the concepts of data sets with an additional dimension of “rate of change”.

These size of data sets varies for each environment as every environment is unique. For example, the rate of change of an environment with a considerable amount of property-result data impacts the database differently than the one that has a sizeable amount of fixlet data (property results vary in data return, but fixlet data is binary). The varying width of data is substantial between these two data types and impacts the size also. Sizing considerations comes always depend on numerous factors. Because of the dynamic nature of data sources and the number of objects ingested and linked to drives the initial full ETL data set, and the rate of change drives the subsequent sets of changes. Consequently, the approach to sizing focuses primarily on the size of the first full data set. The sets of changes are accommodated as a rate of change (percentage) that is applied as an assumption to the full data set. That assumption drives the projected rate of database growth over subsequent ETL changes. In concept, consider the following categories:

Name	Description
Category 1 ETL (Initial)	Provides the Initial reporting dataset. This dataset has the most impact on the database size over time.

Name	Description
Category 2 ETL (point in time and change)	Provides a historical or point-in-time data set. This data set only retrieves the values that have changed since the last ETL. This ETL type is driven by the rate of change in the sourcing data sources. If the rate of change is minimal, the impact of growth is minimal in addition to the time that is required to process an ETL. However, if the rate of change is significant, storage resources and the ETL completion time are affected.

Some general trends correlate the initial data source size to the resulting Insights database size (after the first ETL). This resulting size can vary. However, the following query provides a rough sizing estimate of the database containers that Insights requires. Run the attached query on the ingesting BigFix Enterprise Database. The returned number is a conservative estimate of what the initial dataset might be after a 20% increase. Ensure that you run the query on all ingesting BigFix Enterprise environments to project the accumulated size of the first ETL.

### Query-FullRun estimation

```
-- SCRIPT USED to Gueestimate the destination size database +- 20% For Full
run

Select 1.2* (sum(TotalspaceMB)) as 'Initial Projected DB Container space
required (MB)' from
(SELECT
t.NAME AS TableName,
--s.Name AS SchemaName, p.rows,
CAST(ROUND(((SUM(a.total_pages) * 8) / 1024.00), 2) AS NUMERIC(36, 2))
AS TotalSpaceMB,
CAST(ROUND(((SUM(a.used_pages) * 8) / 1024.00), 2) AS NUMERIC(36, 2))
AS UsedSpaceMB,
```

```

CAST(ROUND(((SUM(a.total_pages) - SUM(a.used_pages)) * 8) / 1024.00, 2) AS
  NUMERIC(36, 2)) AS UnusedSpaceMB
FROM
sys.tables t INNER JOIN
sys.indexes i ON t.OBJECT_ID = i.object_id INNER JOIN
sys.partitions p ON i.object_id = p.OBJECT_ID AND i.index_id = p.index_id
INNER JOIN
sys.allocation_units a ON p.partition_id = a.container_id LEFT OUTER JOIN
sys.schemas s ON t.schema_id = s.schema_id WHERE
--t.NAME NOT LIKE 'dt%'
t.is_ms_shipped = 0 AND i.OBJECT_ID > 255
and (
t.name like 'ACTIONRESULTS' OR t.name like 'ACTIONS'
OR t.name like 'ACTIONSTATESTRINGS'
OR t.name like 'COMPUTERS'
OR t.name like 'COMPUTER_GROUPS'
OR t.name like 'COMPUTER_SITES' OR t.name like 'CUSTOM_ANALYSES'
OR t.name like 'CUSTOM_ANALYSIS_PROPERTIES'
OR t.name like 'CUSTOM_FIXLETS'
OR t.name like 'CUSTOM_FIXLET_FIELDS' OR t.name like 'DBINFO'
OR t.name like 'EXTERNAL_ANALYSES'
OR t.name like 'EXTERNAL_ANALYSIS_PROPERTIES'
OR t.name like 'EXTERNAL_FIXLETS'
OR t.name like 'EXTERNAL_FIXLET_FIELDS' OR t.name like 'FIXLETRESULTS'
OR t.name like 'GROUPS'
OR t.name like 'LONGQUESTIONRESULTS'
OR t.name like 'PROPERTIES'
OR t.name like 'QUESTIONRESULTS' OR t.name like 'SITENAMEMAP'
OR t.name like 'SITES'
) GROUP BY
t.Name, s.Name, p.Rows
)

```

```

as T1
-- SCRIPT USED to Guestimate the destination size T-Log Container +- 20%
For Full run
Select 2.5*(sum(TotalspaceMB)) as 'Initial Projected T-Log Container space
required (MB)' from
(SELECT t.NAME AS TableName,
--s.Name AS SchemaName, p.rows,
CAST(ROUND(((SUM(a.total_pages) * 8) / 1024.00), 2) AS NUMERIC(36, 2))
AS TotalSpaceMB,
CAST(ROUND(((SUM(a.used_pages) * 8) / 1024.00), 2) AS NUMERIC(36, 2))
AS UsedSpaceMB,
CAST(ROUND(((SUM(a.total_pages) - SUM(a.used_pages)) * 8) / 1024.00, 2) AS
NUMERIC(36, 2)) AS UnusedSpaceMB
FROM sys.tables t INNER JOIN
sys.indexes i ON t.OBJECT_ID = i.object_id INNER JOIN
sys.partitions p ON i.object_id = p.OBJECT_ID AND i.index_id = p.index_id
INNER JOIN
sys.allocation_units a ON p.partition_id = a.container_id LEFT OUTER JOIN
sys.schemas s ON t.schema_id = s.schema_id WHERE
--t.NAME NOT LIKE 'dt%'
t.is_ms_shipped = 0 AND i.OBJECT_ID > 255
and (
t.name like 'ACTIONRESULTS' OR t.name like 'ACTIONS'
OR t.name like 'ACTIONSTATESTRINGS'
OR t.name like 'COMPUTERS'
OR t.name like 'COMPUTER_GROUPS' OR t.name like 'COMPUTER_SITES'
OR t.name like 'CUSTOM_ANALYSES'
OR t.name like 'CUSTOM_ANALYSIS_PROPERTIES'
OR t.name like 'CUSTOM_FIXLETS'
OR t.name like 'CUSTOM_FIXLET_FIELDS' OR t.name like 'DBINFO'
OR t.name like 'EXTERNAL_ANALYSES'
OR t.name like 'EXTERNAL_ANALYSIS_PROPERTIES'

```

```

OR t.name like 'EXTERNAL_FIXLETS'
OR t.name like 'EXTERNAL_FIXLET_FIELDS' OR t.name like 'FIXLETRESULTS'
OR t.name like 'GROUPS'
OR t.name like 'LONGQUESTIONRESULTS'
OR t.name like 'PROPERTIES'
OR t.name like 'QUESTIONRESULTS' OR t.name like 'SITENAMEMAP'
OR t.name like 'SITES'
) GROUP BY
t.Name, s.Name, p.Rows
)
as T1

-- SCRIPT USED to Guestimate the TempDB database +- 20% For Full run
Select .5*(sum(TotalspaceMB)) as 'Initial Projected TempDB space required
(MB)' from
(SELECT t.NAME AS TableName,
--s.Name AS SchemaName,
p.rows,
CAST(ROUND(((SUM(a.total_pages) * 8) / 1024.00), 2) AS NUMERIC(36, 2))
AS TotalSpaceMB,
CAST(ROUND(((SUM(a.used_pages) * 8) / 1024.00), 2) AS NUMERIC(36, 2))
AS UsedSpaceMB,
CAST(ROUND(((SUM(a.total_pages) - SUM(a.used_pages)) * 8) / 1024.00, 2) AS
NUMERIC(36, 2)) AS UnusedSpaceMB
FROM
sys.tables t INNER JOIN
sys.indexes i ON t.OBJECT_ID = i.object_id INNER JOIN
sys.partitions p ON i.object_id = p.OBJECT_ID AND i.index_id = p.index_id
INNER JOIN
sys.allocation_units a ON p.partition_id = a.container_id LEFT OUTER JOIN
sys.schemas s ON t.schema_id = s.schema_id WHERE
--t.NAME NOT LIKE 'dt%'

```

```

t.is_ms_shipped = 0 AND i.OBJECT_ID > 255
and (
t.name like 'ACTIONRESULTS' OR t.name like 'ACTIONS'
OR t.name like 'ACTIONSTATESTRINGS'
OR t.name like 'COMPUTERS'
OR t.name like 'COMPUTER_GROUPS' OR t.name like 'COMPUTER_SITES' OR t.name
  like 'CUSTOM_ANALYSES'
OR t.name like 'CUSTOM_ANALYSIS_PROPERTIES'
OR t.name like 'CUSTOM_FIXLETS'
OR t.name like 'CUSTOM_FIXLET_FIELDS' OR t.name like 'DBINFO'
OR t.name like 'EXTERNAL_ANALYSES'
OR t.name like 'EXTERNAL_ANALYSIS_PROPERTIES'
OR t.name like 'EXTERNAL_FIXLETS'
OR t.name like 'EXTERNAL_FIXLET_FIELDS' OR t.name like 'FIXLETRESULTS'
OR t.name like 'GROUPS'
OR t.name like 'LONGQUESTIONRESULTS'
OR t.name like 'PROPERTIES'
OR t.name like 'QUESTIONRESULTS' OR t.name like 'SITENAMEMAP'
OR t.name like 'SITES'
) GROUP BY
t.Name, s.Name, p.Rows
)
as T1

```

## Scale Implications, Best practices and Optimizations

To minimize the associated impact of data over time and optimize the ETL process time, the following principles can be considered for good results:

- Reduce the amount of sites that are ingested (dialog provided per dataset). If you no plans to report on the data history, do not drive the data into Insights.
- After a primary site is added, that data is maintained historically and the data cannot be deleted. Plan the sites you want to import during the initial BigFix Insights setup.

- Plan to run the ETL at a frequency that is useful to business operations. The suggested frequency is once a day.
- Reduce checklist site subscriptions to computers that you intend to evaluate.

## SQL transactions and application behavioral patterns

All transactions use snapshot isolation during importation. From that point, SQL uses a prepared batch statement, and this statement is fed into an upsert. The upsert performs two functions: First is it reads the applicable destination row to see whether an update is required and the second is it inserts or updates the row as required. During importation, the tempdb is heavily used. Data is dumped into the tempdb and then read and processed for updation or insertion.

The end of the importation process, where global objects are computed and stored (warehousing), includes a large read impact (approximately 90%) and small write impact (approximately 10%).

Normal Business Intelligence report running activities generates a large read impact and nearly no write operations.

Considering the previous information, the suggested balance for the Insights DBMS should be optimized around 75 to 25 percent with a slight preference for read based off application usage. For this reason the following specifications are defined.

## BigFix Insights SQL configuration guidelines

Configuration	Description or rationale
Make the tempdb I/O channel dedicated and not shared.	Insights leverages the tempdb Heavily, Isolate TempDB when possible from other workloads.
Configure the SQL memory limitations correctly.	Ensure that SQL memory is capped to allow at least 8 GB for the operating system. This cap is configured in the SQL Server properties.

Configure virus scanners to exclude the SQL file storage location, including all data file sets.	When you configure antivirus software settings, exclude the following files or directories (as applicable) from virus scanning. This exclusion improves the performance of the files and make sure that the files are not locked when the SQL Server service must use them. Refer to instructions link from your virus scanner for more information on how to set this exclusion rule. For information about choosing antivirus software, see <a href="#">How to choose antivirus software to run on computers that are running SQL Server</a>
Do not use file indexing or file compression on supporting SQL data files.	Apply the reasoning that is similar to the reasoning behind antivirus file exclusions to HIPS (Host Intrusion Prevention) based applications or file indexing operations that might lock the data files that are in use.
Initially, establish the sizes of the transaction log and the MDF file to 80 percent of the size projection, and set auto growth appropriately.	By using the provided sizing information, make sure the MDF and LDF data files are initially established prior to setting up the initial database. With this setup, the system can minimize auto growth during the initial ETL.
Configure auto growth of SQL Database files to be substantial versus minimal.	The supporting DB file can become quite large in SQL in support of the Insights database. To minimize the time and resources that the system dedicates to the growing supporting data files, assure the growth characteristics are altered from the default settings for auto-growth to be no less than 2GB or by 10 per every time. In most cases the percentage approach yields the best results.
Ensure that Soft-NUMA is in place.	According to Microsoft, SQL 2017 and 2019 do not use Soft-NUMA and manages NUMA configurations by default. Ensure that this set-



ting is altered from the default setting. For details, refer to [Soft-NU-MA \(SQL Server\)](#)

## ETL rhythm and scheduling

You can control the ETL schedule through Insight administrators. The less frequently ETLs run, the duration to run the ETL is longer because of the increased rate of change. Conversely, running ETLs more frequently likely decreases the duration of an ETL run. However, more frequent runs might consume more space because more trend data is recorded. The correlation of duration and frequency versus size and storage requirements is not necessarily predictable by a single equation and varies greatly based on the BigFix Enterprise Server environment and business practices and conditions. For best results, after you run the initial ETL, attempt to run an ETL every day over the same constant data set (based upon filtered sites). Then, daily measure the increment to the database size daily and the ETL duration, which you can view on the import screen. This practice, can help you to define the rate of change over time. You can use this metric to determine projected database size based upon calculated ETL frequencies. Then, data is captured and recorded. You can further understand the balance of storage requirements versus ETL frequency and better calculate the ETL duration.



**Note:** Although ETLs can be scheduled concurrently, the initial release queues ETLs from data sources. Consider this scenario when you define ETL schedules.

## Managing the replica

BigFix Insights relies on a replica of the BigFix Enterprise Data Source that is co-located with the Insights database. The WebUI Component of Insights is located on a designated WebUI Server, which is likely one of the ingested BigFix Enterprise Servers. The BigFix Insights database can also be co-located on the same WebUI Server. No technical controls prevent you from the putting the database on the WebUI Server. The WebUI Server maintains communication process to the BigFix Insights Database, while all ingested information and audit details are stored on the Insights database. The Insights database might consume more than one BigFix Enterprise Database, and if co-located on the WebUI Server, it might

expand the information boundary to other BigFix Enterprise Data Sources on the Insights Database.

### Backup approach

You must have backups of existing BigFix Enterprise Databases to reduce potential downtime and improve recovery intervals. You can use automation to periodically backup the BigFix Enterprise Databases from the primary BigFix Server and restore backup sets on another SQL server as a snapshot in time and replica. You can generate a snapshot and replica a number of ways, depending on the available tools and utilities.

The following list describes one backup method that uses common tool sets:

- BigFix Server: - Periodic database backup (BigFix Enterprise and BigFix Enterprise Server Reporting).
- Primary BigFix Server: Periodic scheduled task. Generate Robocopy Database backups and BigFix Enterprise Server backups to standby server.
- BigFix replica server: Periodic database restore (BigFix Enterprise and BigFix Enterprise Server Reporting).

### BigFix Server - Periodic Database backup (*BFEnterprise*)



**Note:** The following sample scripts may require editing to run in your environment.

```
USE [msdb]
GO
/***** Object: Job [BigFix Database Backups] 13:58:51 *****/
BEGIN TRANSACTION
DECLARE @ReturnCode INT SELECT @ReturnCode = 0
/***** Object: JobCategory [Database Maintenance] Script Date: 11/16/2012
13:58:51 *****/
IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE
name=N'Database Maintenance'
AND category_class=1) BEGIN EXEC @ReturnCode = msdb.dbo.sp_add_category
@class=N'JOB',
```

```

@type=N'LOCAL', @name=N'Database Maintenance'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
END
DECLARE @jobId BINARY(16)
EXEC @ReturnCode = msdb.dbo.sp_add_job @job_name=N'BigFix Database
Backups',
@enabled=1,
@notify_level_eventlog=0, @notify_level_email=0, @notify_level_netsend=0,
@notify_level_page=0, @delete_level=0,
@description=N'No description available.', @category_name=N'Database
Maintenance',
@owner_login_name=N'BIGFIXVM\Administrator', @job_id = @jobId OUTPUT IF
(@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [Backup BESReporting] Script Date: 11/16/2012 13:58:52
*****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId,
@step_name=N'Backup BESReporting',
@step_id=1, @cmdexec_success_code=0, @on_success_action=3,
@on_success_step_id=0, @on_fail_action=3, @on_fail_step_id=0,
@retry_attempts=0, @retry_interval=0,
@os_run_priority=0, @subsystem=N'TSQL',
@command=N'-----
-
-- Backup database to file
declare @backupFileName varchar(100), @backupDirectory varchar(100),
@databaseDataFilename varchar(100), @databaseLogFilename varchar(100),
@databaseDataFile varchar(100), @databaseLogFile varchar(100),
@databaseName varchar(100), @execSql varchar(1000)

-- Set the name of the database to backup
set @databaseName = 'BESReporting'

```

```

-- Set the path fo the backup directory on the sql server pc set
@backupDirectory =
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Backup\'
-- such as 'c:\temp\'
-- Create the backup file name based on the backup directory, the database
name and today's date
set @backupFileName = @backupDirectory + @databaseName + '-' +
replace(convert(varchar, getdate(), 110), '-', ''), '.') + '.bak'

-- Get the data file and its path
select @databaseDataFile = rtrim([Name]), @databaseDataFilename =
rtrim([Filename])
from master.dbo.sysaltfiles as files inner join
master.dbo.sysfilegroups as groups on
files.groupID = groups.groupID where DBID = (
select dbid
from master.dbo.sysdatabases where [Name] = @databaseName
)

-- Get the log file and its path
select @databaseLogFile = rtrim([Name]), @databaseLogFilename =
rtrim([Filename])
from master.dbo.sysaltfiles as files where DBID = (
select dbid
from master.dbo.sysdatabases
where [Name] = @databaseName
)
and groupID = 0 set @execSql = '
backup database [' + @databaseName + '] to disk = '''' +
@backupFileName + '''' with
noformat, noinit, name = '''' + @databaseName + ' backup''', skip,
compression'

```

```

exec(@execSql)', @database_name=N'master', @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [Backup BFEnterprise] Script Date: 11/16/2012 13:58:52
*****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId,
    @step_name=N'Backup BFEnterprise',
@step_id=2, @cmdexec_success_code=0, @on_success_action=1,
    @on_success_step_id=0, @on_fail_action=2, @on_fail_step_id=0,
    @retry_attempts=0, @retry_interval=0,
@os_run_priority=0, @subsystem=N'TSQL',
@command=N'-----
-
-- Backup database to file
declare @backupFileName varchar(100), @backupDirectory varchar(100),
    @databaseDataFilename varchar(100), @databaseLogFilename varchar(100),
    @databaseDataFile varchar(100), @databaseLogFile varchar(100),
    @databaseName varchar(100), @execSql varchar(1000)

-- Set the name of the database to backup set @databaseName =
    'BFEnterprise'

-- Set the path fo the backup directory on the sql server pc set
@backupDirectory =
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Backup\'
-- such as 'c:\temp\'

-- Create the backup file name based on the backup directory, the database
name and today's date
set @backupFileName=@backupDirectory + @databaseName+'-' +
    replace(convert(varchar, getdate(), 110), '-', ''), ''.'+'.bak'

-- Get the data file and its path
select @databaseDataFile = rtrim([Name]), @databaseDataFilename =
    rtrim([Filename])

```

```

from master.dbo.sysaltfiles as files inner join
master.dbo.sysfilegroups as groups on
files.groupID = groups.groupID where DBID = (
select dbid
from master.dbo.sysdatabases
where [Name] = @databaseName
)

-- Get the log file and its path
select @databaseLogFile = rtrim([Name]), @databaseLogFilename =
rtrim([Filename])
from master.dbo.sysaltfiles as files where DBID = (
select dbid
from master.dbo.sysdatabases where [Name] = @databaseName
)
and
groupID = 0

set @execSql = ''
backup database [' + @databaseName + '] to disk = '''''' +
@backupFileName + '''''' with
noformat, noinit,
name = '''''' + @databaseName + ' backup''', skip,
compression''

exec(@execSql)', @database_name=N'master', @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId, @start_step_id
= 1
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

```

```

EXEC @ReturnCode = msdb.dbo.sp_add_jobschedule @job_id=@jobId,
    @name=N'Backup Schedule',
@enabled=1, @freq_type=4, @freq_interval=1, @freq_subday_type=1,
    @freq_subday_interval=0,
@freq_relative_interval=0, @freq_recurrence_factor=0,
    @active_start_date=20121113, @active_end_date=99991231,
    @active_start_time=20000, @active_end_time=235959,
@schedule_uid=N'16b08a53-4276-4cf4-8d40-d33ff7794db5'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId, @server_name
= N'(local)'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback COMMIT
TRANSACTION
GOTO EndSave QuitWithRollback:
IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION
EndSave:
GO

```

## Primary BigFix Server – Periodic Scheduled Task: Robocopy Database and BES Server backups to Standby Server



**Note:** These commands are run from a Command Prompt or PowerShell environment.

```

forfiles -p "C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER
\MSSQL\Backup" -m *.* -d -7 -c "cmd /C echo @path" >>
C:\maintenance\deleted.txt
forfiles -p "C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER
\MSSQL\Backup"
-m *.* -d -7 -c "cmd /c del /q @path"
robocopy "C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL
\Backup"

```

```
"\\Replica\c$\dbbackups" /s /z /np /w:1 /r:1 /mir /log:C:\Maintenance
\robocopylog.txt /TEE /maxage:3
```

## BigFix replica Server – Periodic Database restore (*BFEnterprise*)



**Note:** The following sample scripts may require editing to run in your environment.

```
USE [msdb]
GO

/***** Object: Job [Restore BigFix Databases] Script Date: 11/16/2012
14:07:56 *****/
BEGIN TRANSACTION
DECLARE @ReturnCode INT SELECT @ReturnCode = 0
/***** Object: JobCategory [Database Maintenance] Script Date:
11/16/2012 14:08:32 *****/
IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE
name=N'Database Maintenance' AND category_class=1)
BEGIN
EXEC @ReturnCode = msdb.dbo.sp_add_category @class=N'JOB', @type=N'LOCAL',
@name=N'Database Maintenance'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
END
DECLARE @jobId BINARY(16)
EXEC @ReturnCode = msdb.dbo.sp_add_job @job_name=N'Restore BigFix
Databases',
@enabled=1, @notify_level_eventlog=0, @notify_level_email=0,
@notify_level_netsend=0, @notify_level_page=0, @delete_level=0,
@description=N'No description available.', @category_name=N'Database
Maintenance', @owner_login_name=N'DSSSUA\Administrator', @job_id = @jobId
OUTPUT
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
```



```

/***** Object: Step [Restore BESReporting] Script Date: 11/16/2012
14:10:28 *****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId,
    @step_name=N'Restore BESReporting',
@step_id=1, @cmdexec_success_code=0, @on_success_action=3,
    @on_success_step_id=0, @on_fail_action=3, @on_fail_step_id=0,
    @retry_attempts=0, @retry_interval=0,
@os_run_priority=0, @subsystem=N'TSQL',
@command=N'-----
-
-- Restore database from file
use master go
declare @backupFileName varchar(100), @restoreDirectory varchar(100),
    @databaseDataFilename varchar(100), @databaseLogFilename varchar(100),
@databaseDataFile varchar(100), @databaseLogFile varchar(100),
    @databaseName varchar(100), @execSql nvarchar(1000)

-- Set the name of the database to restore set @databaseName =
    'BESReporting'
-- Set the path to the directory containing the database backup set
@restoreDirectory = 'C:\DBBackups\' -- such as 'c:\temp\'

-- Create the backup file name based on the restore directory, the database
name and today's date
set @backupFileName=@restoreDirectory+@databaseName + '-' +
    replace(convert(varchar, getdate(), 110), '-', '.') + '.bak'

-- Get the data file and its path
select @databaseDataFile = rtrim([Name]), @databaseDataFilename =
    rtrim([Filename])
from master.dbo.sysaltfiles as files inner join
master.dbo.sysfilegroups as groups on

```

```

files.groupID = groups.groupID where DBID = (
select dbid
from master.dbo.sysdatabases where [Name] = @databaseName
)

-- Get the log file and its path
select @databaseLogFile = rtrim([Name]), @databaseLogFilename =
    rtrim([Filename])
from master.dbo.sysaltfiles as files where DBID = (
select dbid
from master.dbo.sysdatabases where [Name] = @databaseName
)
and
groupID = 0

-- Create the sql to kill any active database connections set @execSql =
    ''
select @execSql = @execSql + 'kill ' + convert(char(10), spid) + ' '
    from master.dbo.sysprocesses
where db_name(dbid) = @databaseName and
DBID <> 0
and
spid <> @@spid exec (@execSql)

set @execSql = ''
restore database [' + @databaseName + '] from disk = '''''' +
    @backupFileName + '''''' with
file = 1,
move '''''' + @databaseDataFile + '''''' to '' + '''''''''' +
    @databaseDataFilename + ''''''',
move '''''' + @databaseLogFile + '''''' to '' + '''''''''' +
    @databaseLogFilename + ''''''',

```

```

replace''

exec sp_executesql @execSql', @database_name=N'master', @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [Restore BFEnterprise] Script Date: 11/16/2012
14:10:34 *****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId,
    @step_name=N'Restore BFEnterprise',
@step_id=2, @cmdexec_success_code=0, @on_success_action=1,
    @on_success_step_id=0, @on_fail_action=2, @on_fail_step_id=0,
    @retry_attempts=0, @retry_interval=0,
@os_run_priority=0, @subsystem=N'TSQL',

@command=N'-----
-
-- Restore database from file

use master go
declare @backupFileName varchar(100), @restoreDirectory varchar(100),
    @databaseDataFilename varchar(100), @databaseLogFilename varchar(100),
    @databaseDataFile varchar(100), @databaseLogFile varchar(100),
    @databaseName varchar(100), @execSql nvarchar(1000)
-- Set the name of the database to restore set @databaseName =
    'BFEnterprise'
-- Set the path to the directory containing the database backup set
    @restoreDirectory = 'C:\DBBackups\' -- such as 'c:\temp\'
-- Create the backup file name based on the restore directory, the database
    name and today's date
set @backupFileName = @restoreDirectory + @databaseName + '-'' +
    replace(convert(varchar, getdate(), 110), '-', ''), ''.'') + '.bak'

-- Get the data file and its path

```

```

select @databaseDataFile = rtrim([Name]), @databaseDataFilename =
    rtrim([Filename])
from master.dbo.sysaltfiles as files inner join
master.dbo.sysfilegroups as groups on
files.groupID = groups.groupID where DBID = (
select dbid
from master.dbo.sysdatabases where [Name] = @databaseName
)

-- Get the log file and its path
select @databaseLogFile = rtrim([Name]), @databaseLogFilename =
    rtrim([Filename])
from master.dbo.sysaltfiles as files where DBID = (
select dbid
from master.dbo.sysdatabases where [Name] = @databaseName
)
and
groupID = 0

-- Create the sql to kill any active database connections
set @execSql = ''
select @execSql = @execSql + 'kill ' + convert(char(10), spid) + ' '
    from master.dbo.sysprocesses
where db_name(dbid) = @databaseName and
DBID <> 0
and
spid <> @@spid exec (@execSql)

set @execSql = ''
restore database [' + @databaseName + '] from disk = '''''' +
    @backupFileName + '''''' with
file = 1,

```

```

move ' ' + @databaseDataFile + ' ' to ' ' + ' ' +
  @databaseDataFilename + ' ',
move ' ' + @databaseLogFile + ' ' to ' ' + ' ' +
  @databaseLogFilename + ' ',
replace''

exec sp_executesql @execSql', @database_name=N'master', @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId, @start_step_id
= 1
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback EXEC
  @ReturnCode = msdb.dbo.sp_add_jobschedule @job_id=@jobId,
@name=N'Restore Schedule', @enabled=1,
@freq_type=4,
@freq_interval=1, @freq_subday_type=1, @freq_subday_interval=0,
  @freq_relative_interval=0, @freq_recurrence_factor=0,
  @active_start_date=20121113, @active_end_date=99991231,
  @active_start_time=50000, @active_end_time=235959,
@schedule_uid=N'1fd12e1c-1a21-49d3-b57a-e8195c980b29'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId, @server_name
= N'(local)'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback COMMIT
TRANSACTION
GOTO EndSave QuitWithRollback:
IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION
EndSave:
GO

```

## Appending SQL and SAN

### SAN considerations

You have to consider several factors when you install and configure an SQL Server to a SAN based environment. You must be familiar with both SQL and SAN.

Make sure that you familiarize yourself with the fundamentals of SQL Servers, when you install a SAN. Whether or not a SAN is being used, it is important to have a good knowledge of the hardware that SQL files are being stored on. The configuration values must not conflict with the fundamentals.

- All SAN hardware must comply with SQL Server I/O basics. The write ordering needs must be maintained, which means that writes must always be secured in or on stable media and you must take precautions to prevent torn I/O write actions. The write ordering process must be available from the selected manufacturer or the manufacturer's SAN documentation.

1. Stable media refers to storage that can survive system restart or common failure and is physical disk storage, but SAN caches also fall under this definition. Stable media is an SQL Server requirement, and the server relies on stable media for Write Ahead Logging to maintain the ACID properties of the database and to ensure data integrity. Stable media strategies must be supported by the hardware manufacturer.
2. Write ordering is the ability to preserve the order of write operations and is a requirement of SQL Server on Hardware. SQL Server requires write ordering to be maintained for local and remote I/O targets. If write ordering is not maintained, the system breaks the write ahead logging protocol.
3. Torn I/O refers to a situation where a partial write takes place, leaving the data in an invalid state. SQL Server documents this as a "Torn Page." Torn page or I/O prevention means that the disk system must provide sector alignment and sizing in a way that prevents torn I/O, including splitting I/O among various I/O entities in the I/O path.

### **SAN and RAID configurations**

SAN hardware supports zoning, which an admin can use to dedicate full physical disks to a LUN so that the I/O is isolated from the SQL Server. SQL Server performance depends on read/write access to physical disks. So, basic RAID fundamentals are important

when you deploy servers on a SAN. SQL is an I/O intensive application, and the highest priority for performance should be given to the SQL Servers disk configurations. Microsoft recommends to separate SQL log files from data files at a physical disk level. In a SAN environment, this means separating the database log files to LUNs, which have dedicated physical disks, from those that are not shared with other LUNs, including those that are used for other database logs, data, or index files. A write response time of 2 - 3 ms and a read response time no greater than 20ms is the goal for this configuration according to Microsoft best practices. Isolate physical disks that servers or applications share, because different servers that run different applications with different I/O characteristics might cause problems.

### **Further considerations**

On large database servers that use SANs, isolate the TempDB, Logs, and database file subsystems from the physical disk space requirements. Because large database engines usually use TempDBs and transaction logs, additional consideration must be given to separate an SQL TempDB or transaction log files to its own RAID subsystem, preferably RAID 10. The Microsoft SQL best practice typically recommends RAID 1+0 for MDF and Index Files, and Microsoft does not recommend RAID5 because of increased read/write times that are associated with it. Based on Microsoft recommendations use a 250Kb stripe size. Never set stripe sizes lower than 64Kb because that setting is the size of a single SQL extent.

### **Storage requirements**

It is an advantage to have distinct storage subsystems and controllers for specific components. For best results, set up discrete storage subsystems for each of the bases, the database logs, and the database containers. The database logs are typically characterized by the need for very fast sequential I/O. The database containers can have much more diverse access patterns, but are highly insulated from storage impact because of the database buffer pools.

## **Known issues and limitations**

### **Insights data lake**

- Datasource authentication is currently limited to SQL authentication.
- Linked objects is limited to external sites, Fixlets, tasks, and analysis.



**Note:** Custom sites are not supported.

- BigFix Enterprise Server operators, roles, and unmanaged assets are not ingested by Insights.
- Insights supports ingesting data from eight or fewer BigFix Enterprise databases, depending on performance qualification.

### **Tableau sample reports**

- The performance of the Tableau reports varies when they display data for a larger set of devices.
- You can use Tableau sample reports only on an BigFix Insights database that has completed an ETL. The reports run as though an ETL is scheduled daily.

### **More information**

For more information, see these documentation sets:

- [BigFix Insights](#)
- [BigFix forum](#)



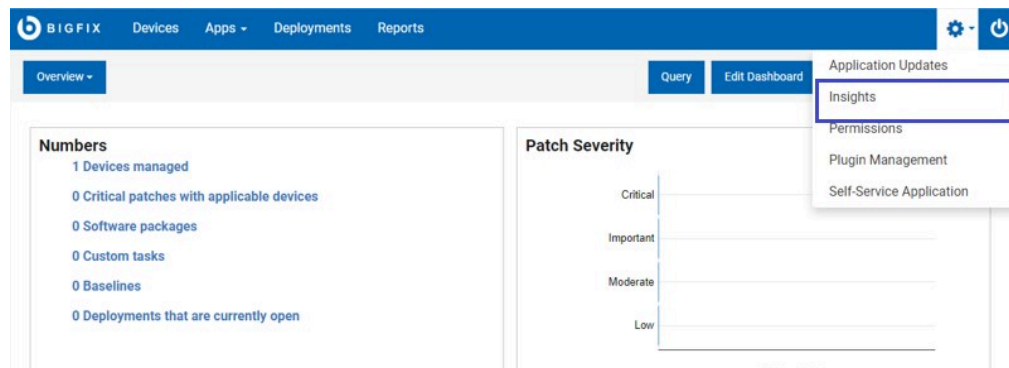
# Chapter 4. Setting up BigFix Insights

Learn how to set up BigFix 10 Insights and edit BigFix 10 Insights connection details.

Initial setup of BigFix Insights is performed within the WebUI app. To install and configure Insights, navigate to the WebUI insights application as a master operator within the deployment.

1. Log in into WebUI by using your Master Operator credentials.
2. Click the gear icon in the navigation bar.

Figure 2. Navigating to Insights



3. Click **Insights**.  
Setup BigFix Insights page is displayed.

Figure 3. Connect to BigFix Insights Database

The screenshot shows the 'Setup BigFix Insights' wizard in the BigFix console. The 'Bigfix Insights Server Connection' section includes a 'Server Alias' field, radio buttons for 'Create New Insights' (selected) and 'Reconnect to Existing Insights', and fields for 'Hostname or IP Address', 'Port' (1433), 'Username', and 'Password'. The 'Authentication' section has 'SQL' selected over 'Windows NT'. The 'Advanced Configuration' section includes a 'Database Instance Name' field, an 'Encryption Enabled' toggle (set to 'Yes'), and a 'Database' field with 'BFInsights' entered. 'Cancel' and 'Next' buttons are at the bottom right.



**Note:** Assure that Create New Insights option is selected.

4. Provide the following connection properties to help define the Insights Datalake's location.

### Server Alias

Name of your Insight's application.



**Note:** The Alias do not correspond to the database name or the server name. It is used to provide a common name to refer to this Insights Datalake instance within the WebUI app. The name of the Alias cannot be changed

### Hostname or IP Address

Target database server that you want the Insights Datalake assigned to.

**Port**

Listening port of the SQL Server that hosts the Insights Datalake (by default SQL listens on 1433).

**Username**

The username that has the permissions to create the Insights Datalake on the target database server that you are connecting to.

**Password**

The password for the associated username.

**Authentication**

Authentication type to authenticate and create the Insights Datalake.

**Advanced Configuration**

Parameters (if required).

**Database Instance Name**

An alternative SQL or Server Instance name.



**Note:** By default SQL leverages a default instance. This configuration is defined during SQL server installation. If no value is entered within this field (default behavior), the default MSSQL instance is targeted to create the Insights Datalake.

**Connection Encrypted**

Define if you want to have the connection Encrypted or not Encrypted.



**Note:** Connection should be Encrypted and this is the default selection.

**Database Name**

The default name of the Insights Datalake is **BFInsights**. If you want to change the default Insights Datalake name from this default database name value, you should define the preferred name.



**Note:** Database Administrators prefer to specify the associated logical volumes for the corresponding database file sets. This is a Microsoft best practice. For example: Transaction logs and database container files should be isolated on separate logical and physical volumes. If the Database Administrator creates the database prior to initiating the Insights installation and the database is named with the same value provided (as it appears within this field), the installation populates the desired database with the Insights schema, and consequently this will inherit the file locations preferred by your DBA. This is only applicable for new installations of the Insights Datalake. Consult your DBA for more information.

#### 5. Click **Next**.

This creates the defined Insights Datalake on the server with the defined connection details. Once you complete this step, the initial database is created.

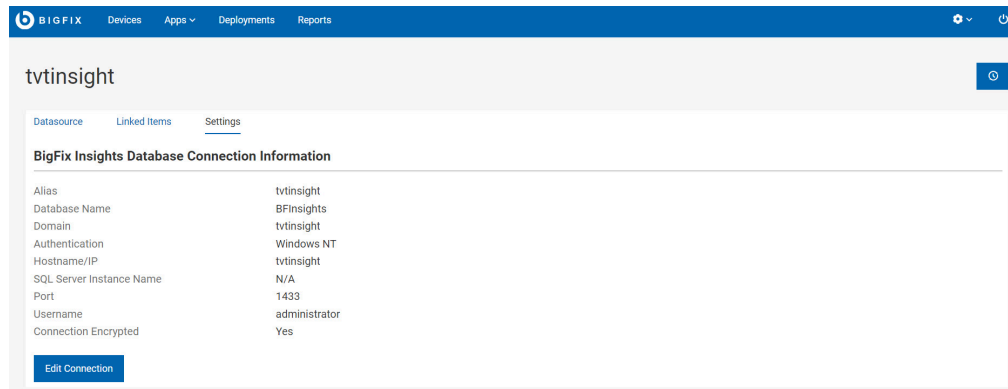
This provisioning of the Insights Datalake is complete. The Insights server homepage is displayed allowing the user to add the initial **BFEnterprise** source to Insights.

Insights requires the target BFE replica details and credentials to ingest the first data source. Refer to *Working with datasources and sites* for configuring a data source.

## Editing BigFix Insights connection details

If you need to change the Insights Datalake location or connection information or alter the Insights credential (including changing the password), use the **Edit Connection** button in the Settings tab of the Insights landing page. This page allows you to see the current connection information and change the connection information. Execute the following steps to view or edit the Insights Datalake connection details.

1. In the Insights server home page, click **Settings** tab to view the connection details.



2. Click **Edit Connection** to change or edit the connection details or credentials leveraged for authentication.



**Note:** The functions facilitated by this page should be understood prior to editing the connection details. This page controls the configuration parameters where the Insights DB is located and incorrect connection details may have an adverse impact. The fields and their purposes are outlined in step 4 of [Setting up BigFix Insights \(on page 37\)](#).

**Bigfix Insights Server Connection**

Server Alias\*  
tvinsight

Domain\*  
tvinsight

Hostname or IP Address\*  
tvinsight

Port\*  
1433

Username\*  
[Redacted]

Password\*  
[Redacted]

Authentication  
SQL Windows NT

**Advanced Configuration**

Database Instance Name  
[Redacted]

Encryption Enabled  
Yes No

Database  
BFinsights

Cancel Save

3. Click **Save** after editing the values to the desired configuration or click **Cancel** to exit.