

BigFix
BigFix 11 Insights 入門ガイド

Special notice

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

Edition notice

This edition applies to BigFix version 11 and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. バージョン履歴.....	5
Chapter 2. 概要.....	6
Chapter 3. 前提条件と制限.....	8
Chapter 4. BigFix Insights のセットアップ	36
Chapter 5. BigFix Insights のアンインストール.....	41

Chapter 1. バージョン履歴

日付	バージョン	Note
2020 年 09 月 30 日	1.0	初期リリース
2021 年 11 月 30 日	2.0	資料は『はじめに』と『操作ガイド』に分割

Chapter 2. 概要

BigFix Insights はすべての BigFix データを単一のデータ・ウェアハウスに統合するのに役立つアプリケーション主導型の抽出、変換、ロード (ETL) です。

Insights は ETL のデプロイメント間のすべてのデバイスと BigFix データの相関図を自動的に作成します。相関図の自動作成後、ビジネス・インテリジェンス (BI) ツールは Insights のデータベースを参照でき、レポートに必要な分析や視覚化を生成し、リスクを軽減できます。

Insights は、複数の BigFix デプロイメントからのデータをデータ・ウェアハウスに格納、集約する機能を持つ ETL を提供します。このデータ・ウェアハウスから、Tableau、Power BI などのビジネス・インテリジェンス・ツールと分析ツールを使用できます。

フィーチャーと機能:

以下は、BigFix Insights に備わっている機能のリストです。

- BigFix Enterprise データソースからのデータを、レポートの生成に最適化されたデータの集められた単一のデータ・レイクにインポートおよび統合します。
- ETL スケジュールを管理するための WebUI ベースのツールを提供します。
- BI のレポート作成ツールを使用して、さまざまな基準や過去のトレンドごとにグループ化したデータの要約を示すレポートを簡単に作成します。レポートには、以下の領域を対象とした豊富な視覚化セットおよび動的フィルター機能を備えています。
 - パッチ・リズム
 - デバイスのインベントリー
 - OS マイグレーション
 - デプロイメント進行状況

技術的背景

BigFix Enterprise Server (BFEEnterprise) データベースには、Fixlet、分析、コンピューター、サイト、およびプロパティーに関する情報が含まれています。BFEEnterprise データベース内に格納されたデータは、ほぼリアルタイムのプラットフォーム機能をサポートするように最適化された非正規化形式で格納されます。Insights は、BFEEnterprise データベース (レプリカ) から情報を取り込み、このデータを標準の正規化された形式に変換します。この形式は、SQL でのクエリー、レポート作成、および組織全体の重要なデータの視覚化サービスをサポートするように最適化されています。このドキュメントでは、BigFix Insights アプリケーションに関連付けられているスキーマの概要も説明します。

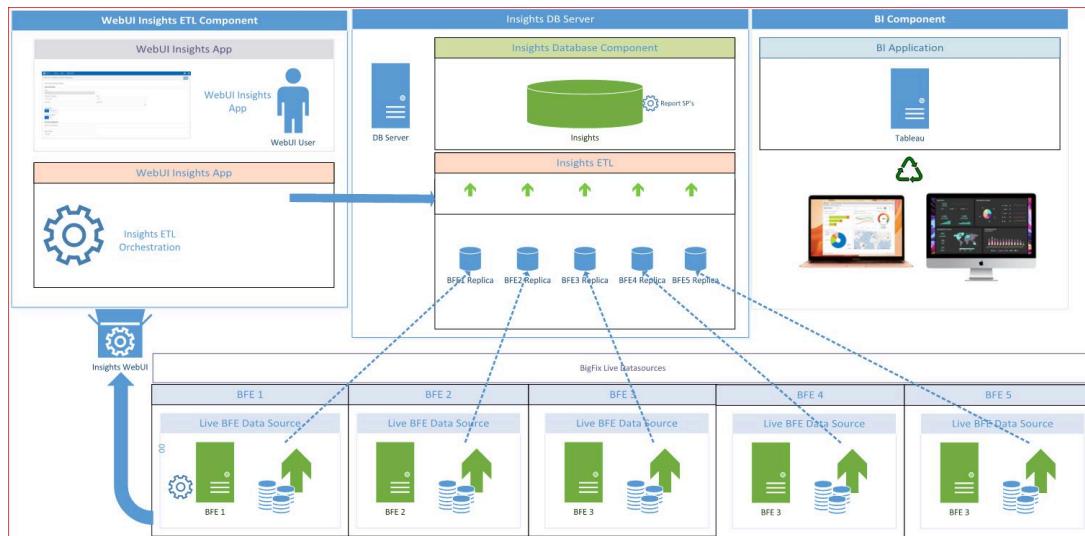
ライセンス基準

BigFix Insights は、すべての既存の BigFix サービス (Patch、Lifecycle、Start Kit、Compliance、または Inventory) のアドオンとして利用可能です。詳細については、お客様の HCL 担当員にお問い合わせください。

アーキテクチャーの概要

下図は、BigFix Insights のアーキテクチャーの概要を示しています。

Figure 1. BigFix Insights - アーキテクチャーの概要



上記のアーキテクチャーは、BigFix Insights のインフラストラクチャー・コンポーネントを示しています。BigFix Insights アプリケーションは MSSQL DBMS に基づいて構築され、WebUI プラットフォームを駆動アプリケーションとして利用します。Insights アプリケーションは、1 つまたは複数の BigFix Enterprise Root Server Databases から構造化データを取り込みます。BigFix Root Server プラットフォームのパフォーマンスを維持し、アプリケーションへの潜在的な影響を軽減するために、データは BFEnterprise データベースのバックアップまたはレプリカ (T-Log 配布を利用) から取得されます。ライブ BFEnterprise データ・ソースでの ETL の実行は、初期リリースではサポートされておらず、すべての実稼働実装シナリオで避ける必要があります。Insights Datalake スキーマはドキュメント化されており、正規化された形式の BFEnterprise データベースを提供するため、お客様は BI ツールを利用してレポートを作成できます。

初期リリースでは、機能の例として、Tableau 内にいくつかのレポートが提供されています。Tableau には豊富なデータ視覚化オプションがあり、レポートの作成と開発のための堅牢なユーザー・インターフェースを提供しますが、お客様は他の BI インターフェースも活用することもできます。

Chapter 3. 前提条件と制限

このセクションでは、BigFix 10 Insights の前提条件と制限に関する詳細情報を提供します。

システムの前提条件

BigFix 10 Insights のシステム要件は次のとおりです。

- BigFix Insights をインストールおよび使用するための使用権。
- BigFix Insights には、BigFix Insights コンテンツ・サイトをサブスクライブしている Windows WebUI サーバーが必要です。



注意:

- Insights は 2 つの BigFix コンテンツ・サイトで構成されます。それは、サンプル・レポートの配信用の外部のコンテンツ・サイトと WebUI コンテンツ・サイトです。
- WebUI コンテンツ・サイトは、ETL を促進する WebUI サーバーにサブスクライブしている必要があります。Tableau サンプル・ワークブックを取得するには、外部のコンテンツ・サイトがコンピューターにサブスクライブしている必要があります。これらのサイトを環境内のすべてのコンピューターにまとめてサブスクライブしてはなりません。
- 現在、Linux WebUI はサポートされていません。
- WebUI の構成については、[を参照してください。\[https://help.hcltechsw.com/bigfix/10.0/webui/WebUI/Admin_Guide/WebUI_admin_guide.html\]\(https://help.hcltechsw.com/bigfix/10.0/webui/WebUI/Admin_Guide/WebUI_admin_guide.html\)](https://help.hcltechsw.com/bigfix/10.0/webui/WebUI/Admin_Guide/WebUI_admin_guide.html)

- BigFix Enterprise Database Ingestion データ・ソースの SQL レプリカまたはオフライン・コピー。



注意:

- 稼働中の Insights ルート・サーバーには BigFix ETL を実行しない。レプリカは、BigFix Insights のデータ・ソースとして機能する必要があります。レプリカは、バックアップ、T-Log 配布、SQL Always On などのさまざまな SQL プロセスによって容易にすることができます。
- Insights ETL は、Windows ベースのルート・サーバー (BigFix 10) からの取り込みのみをサポートします。
- DB2 データベースまたはルート・サーバーはサポートされていません。
- Linux SQL は現在サポートされていません。

- BigFix Insights には、Insights Data Lake 用に Microsoft SQL Server 2017、SQL Server 2019、または SQL Server 2022 が必要です。それより前のバージョンはサポートされていません。
- Insights Data Lake は、WebUI を実行している Windows マシンから生成する必要があります。
- 製品に付属する Tableau レポートを表示するために、提供されている Insights Tableau ワークブックを実行および表示するには、Tableau ライセンス契約が必要です。このライセンスは、Insights とは別の資格です。Tableau Creator ライセンスが 1 つ以上必要です。ワークブックを表示するには、Explorer または View ライセンスを使用できます。

データベースへのアクセス許可

ユーザーには、以下の 3 つのタイプがあります。

1. レポート・ユーザー (PowerBI/Tableau): これらのユーザーには、Insights 内のすべてのテーブルへの SELECT アクセスを許可する READ ONLY 権限が必要です。



Note:

複数のデータソースのシナリオでは、行レベル・セキュリティ (RLS) ポリシーにより、読み取り専用データベース・ユーザーはすべてのテーブル・データにはアクセスできない場合があります。RLS セキュリティ・ポリシーをバイパスしてすべてのデータソースにわたってレポートを生成するには、`dbo` ユーザーを使用する方法と、「`sec_rls_excluded`」というユーザー・グループを作成してユーザーを追加する方法の 2 つがあります。

2. Insights 作成者 (insights_creator) : Insights 作成者には、次の権限が必要です。

- CREATE DATABASE
- ログインの変更
- リンクされたサーバーの変更
- CONNECT SQL
- VIEW SERVER STATE
- INSERT/UPDATE/DELETE/ALTER tempdb

3. Insights 管理者 (insights_Admin) : Insights を使用し、通常の ETL を実行する管理者として、これらのユーザーには次の権限が必要です。

- ログインの変更
- リンクされたサーバーの変更
- CREATE SCHEMA
- dbo への SELECT/INSERT
- WebUI への SELECT/INSERT

- ALTER ANY SECURITY POLICY
- INSERT/UPDATE/DELETE/ALTER tempdb
- VIEW SERVER STATE
- CONNECT SQL

資格情報の前提条件

Insights ETL コンポーネントの資格情報の前提条件は、次のとおりです。

Table 1. 資格情報の前提条件

必要な権限	目的
DB リーダーから取り込むソース (レプリカ)	DB リーダーは、Insights によって取り込まれるデータ・ソースの情報の認証および読み取りに使用されます。
DBO から Insights	Insights アカウントは、進行中または永続的な ETL を使用してアプリケーション・スキーマを維持したり、Insights データベースを作成したりするため使用されます。

ネットワークの前提条件

データをロードするすべてのアプリケーションと同様に、ネットワークの遅延と帯域幅への影響を最小限に抑え、最適化します。可能な場合には、ネットワークに近い場所にリソースを配置します。可能であれば、レプリカを Insights Data Lake と同じ場所に配置します。

Table 2. ネットワーク権限

必要な権限	目的
Insights ETL サーバーは、BigFix Enterprise をサポートする listen 中の SQL ポートを介して、取り込まれたすべての SQL インスタンスまたはサーバー (レプリカ) を明確に把握する必要があります。	Insights Data Lake DB サーバー → 取り込みソース DB: <取り込みソース DB ポート>
WebUI インスタンスを実行しているサーバーは、1433 ポートまたは SQL リスニング・ポート	WebUI サーバー → Insights Data Lake DB サーバー: <取り込みソース DB ポート>

Table 2. ネットワーク権限 (continued)

必要な権限	目的
を使用して Insights ETL サーバー・ データベースに接続する必要があります。	

アプリケーションの要件

DBMS の前提条件

Insights DBMS コンポーネントの DBMS 前提条件は次のとおりです。

Table 3. DBMS の前提条件

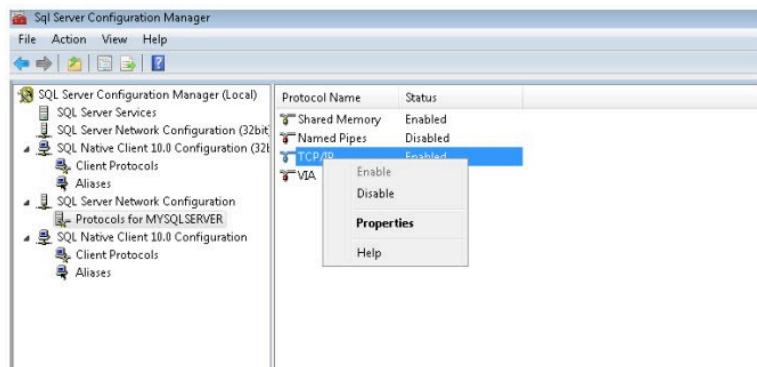
必要な権限	目的
最小取り込みの BigFix Enterprise バージョン	<p>Insights は、1つ以上の BFE データ・ ソースを取り込みます。</p> <ul style="list-style-type: none"> 必要なデータ・ ソースは 9.5 以降である必要があります。 Insights は、DB2 データ・ ソースの取り込みをサポートしていません。
BigFix Enterprise のライブ・ デプロイメントへの Insights の取り込みはサポートされていません。	<ul style="list-style-type: none"> 初期リリースでは、BigFix コア・ プラットフォームへのパフォーマンスへの影響が軽減されました。 Insights は、BigFix Enterprise データベースのオフライン・ コピーまたはレプリカからデータを取り込む必要があります。 <p>Insights データベースのネットワークに近い場所にデータベースのオフライン・ コピーを配置します。</p>

SQL Server リスニング・ ポート

デフォルトでは、SQL Server は TCP ポート番号 1433 で lisiten しますが、名前付きインスタンスの場合は TCP ポートが動的に構成されます。SQL Server という名前付きインスタンスのリスニング・ ポートを取得するために使用できるオプションがいくつかあります。

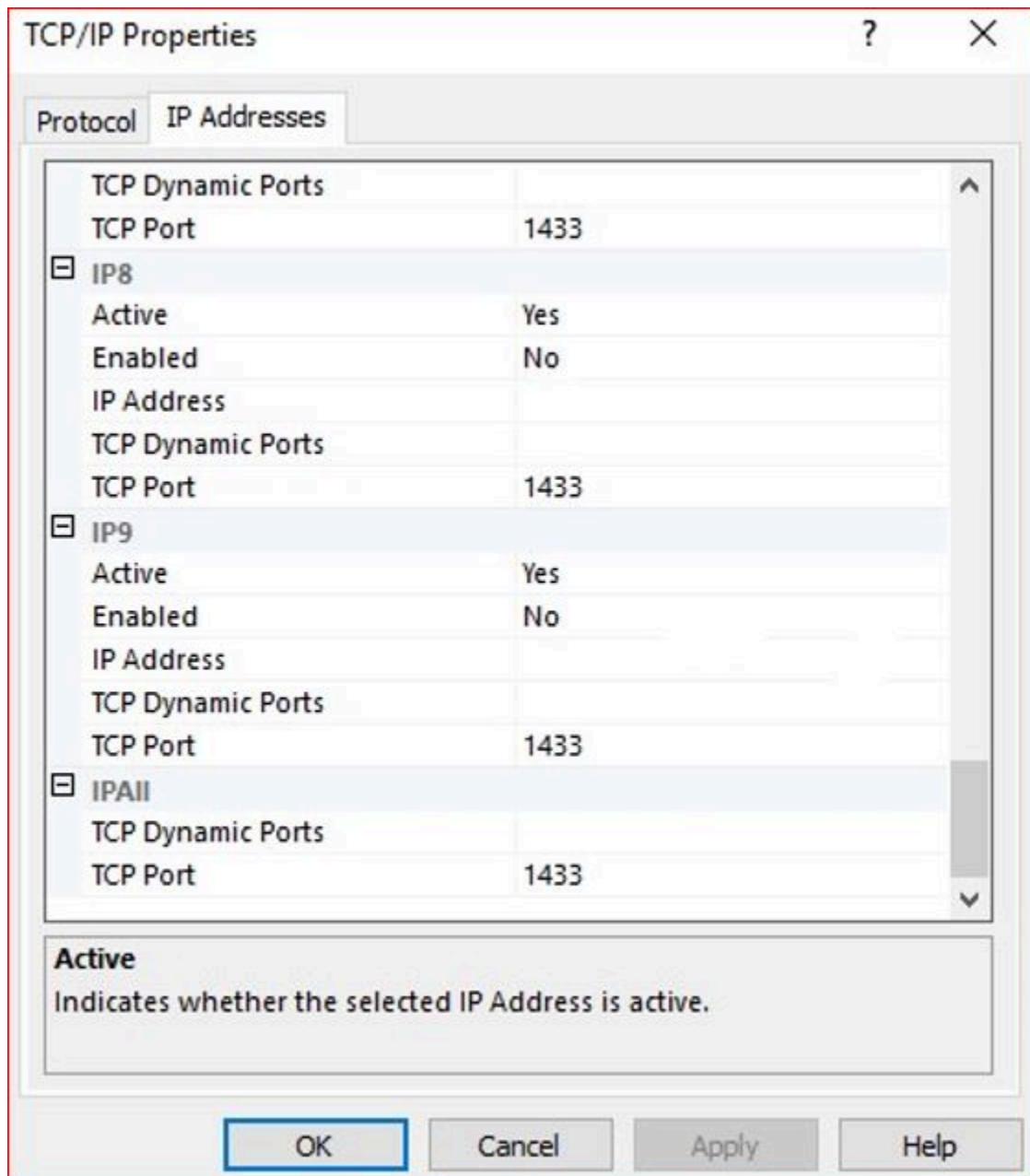
リスニング・ポート情報を取得するには、以下のステップを実行します (SQL Server 構成マネージャー)。

1. 「スタート」 > 「すべてのプログラム」 > **Microsoft SQL Server 2008** > 「構成ツール」 > 「SQL Server 構成マネージャー」の順にクリックします。
2. 「SQL Server 構成マネージャー」 > 「SQL Server ネットワーク構成」 > 「<your Database>のプロトコル」の順に移動します。
3. 「TCP/IP」を右クリックして、「プロパティー」を選択します。



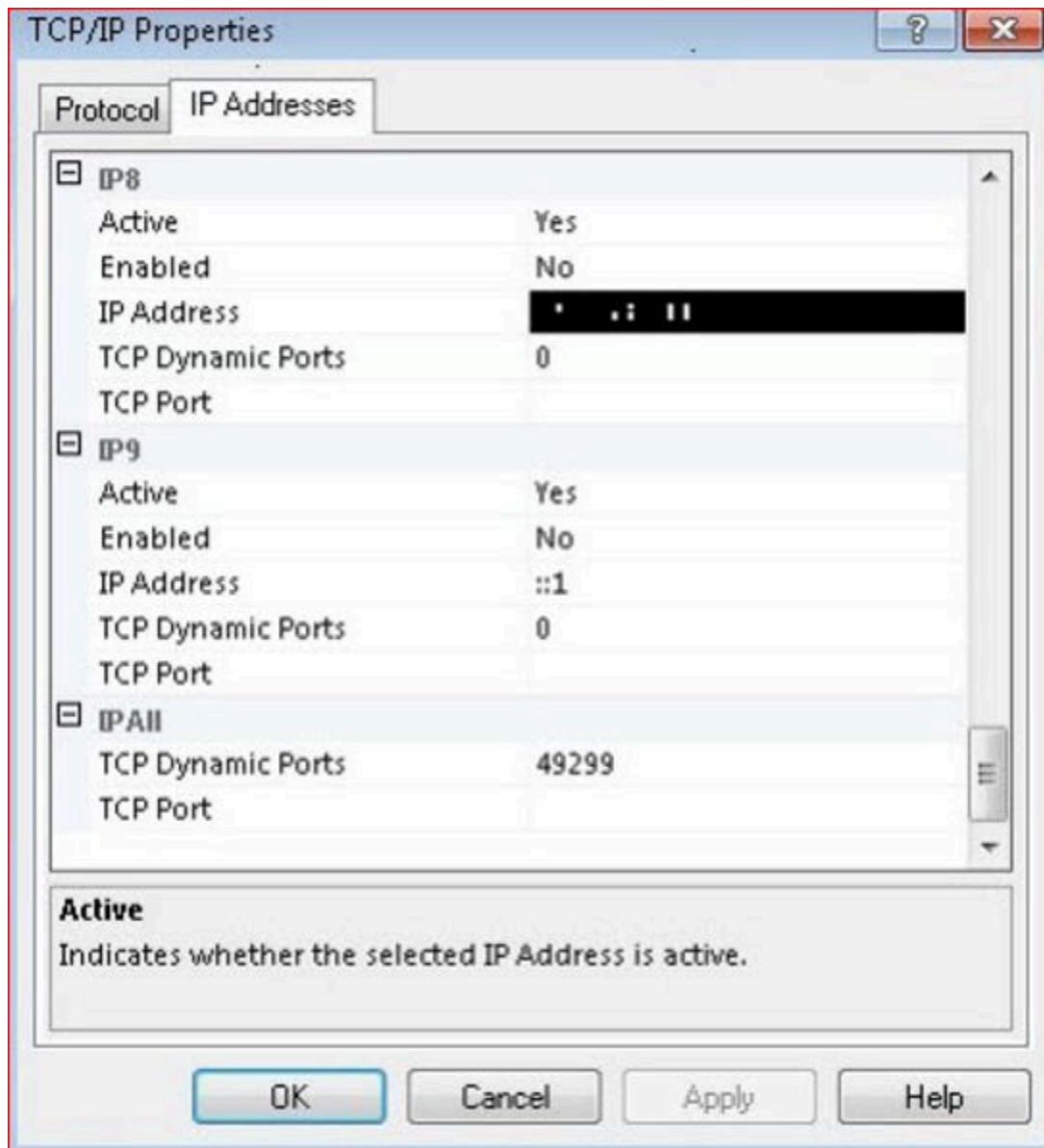
4. 「TCP/IP のプロパティー」ダイアログ・ボックスで、「IP アドレス」タブに移動し、「IPAll」グループまで下にスクロールします。

構成が静的ポート上にある場合は、現在のポート番号が「TCP ポート」テキスト・ボックスに表示されます。



上のスクリーンショットでは、インスタンスはポート番号 1433 で listen しています。

SQL Server が動的ポートで構成されている場合は、現在のポート番号が「TCP 動的ポート」テキスト・ボックスに表示されます。



上のスクリーンショットでは、インスタンスはポート番号 49299 で listen しています。

コンピューターおよびメモリー要件

次のセクションは、『BigFix キャパシティー・プランニング・ガイド』から抜粋したものです。完全なマニュアルを参照してください：[BigFix パフォーマンス & キャパシティー・プランニング・リソースガイド](#)

キャパシティー計画で最善の結果を実現するには、データベース・レプリカに基づいて Insights をデプロイしてください。データベース・バックアップ、MS SQL レプリケーション・サポート、MS SQL 可用性グループの使用など、基本キャパシティーを管理するための多くのオプションが用意されています。

- MS SQL レプリカの場合は、追加の MS SQL リソースが必要です。次の表を参照して、レプリカに必要なログ・スペースに対する追加の負荷を確認してください。
- 小規模デプロイメントやテスト・デプロイメントの場合は、レプリカをバイパスできます。ただし、システムとデータベースの健全性を確保するために、特に DBMS の場合は事前監視と事後監視を使用して、この方法を注意深く使用してください。
- 1 秒当たりの入出力操作数 (IOPS) の場合、一般的な DBMS サーバの基準である 5,000 IOPS と 1 ミリ秒未満の待ち時間が最適な結果を得られます。ただし、I/O プロファイルは多くの読み取りを必要とし、レポートの内容やワークロードの影響を受けやすい傾向があります。その結果、2,500-IOPS ベースのストレージ・デバイスを使用すると、良好なパフォーマンスが得られる可能性があります。ただし、この可能性は慎重な監視を行わないと評価できません。

Insights サーバーの場合、キャパシティー計画の要件は、BigFix の基本製品を対象としています。

- 基本オペレーティング・システム、MS SQL セカンダリー・インスタンス (該当する場合)、および顧客が提供するレポート作成テクノロジー (Tableau など) の要件、および関連するデータ・ソースを含める必要があります。
- 最良の結果を得るために、デプロイメント・サイズは、初期製品向けの BigFix Enterprise コンピューターのテーブル・データ・カーディナリティーで表されます。

CPU とメモリーの推奨事項は、BigFix ルート・サーバーのコンテンツを取り込んで Insights Data Lake を構築する抽出/変換/ロード (ETL) エージェントに基づいています。ETL プロセスの効率化のために、これらの要件は一般的に小さくなります。

Table 4. BigFix Insights MS SQL レプリカのキャパシティー計画要件

コンポーネント	追加 CPU	追加メモリー (GB)	ログ・スペース (GB)
MS SQL レプリケーション	+2	+4	通常のスペースの 2 倍
デプロイメントのサイズ CPU メモリー (GB) ストレージ (GB)			
10,000	+4	+10	BFEnterprise の 10%

ワークロード管理と Insights DBMS のサイズ変更

取り込みオブジェクト

BigFix Insights におけるスケーラビリティの影響を理解するには、BigFix Enterprise データ・ソースからインポートされるオブジェクトを理解する必要があります。このリリースでは、次のオブジェクトがインポートされます。

- アクション
- サイト
- コンピューター
- プロパティー
- プロパティーの結果
- Fixlet (分析、タスク、および Fixlet)、以下で「コンテンツ」と呼ばれる: BigFix Insights
- Fixlet の結果。以下で「コンテンツの結果」と呼ばれる:
- グループ (グループ・メンバーシップを含む)

このリリースでは、Insights をインポートする際に、次のルート・オブジェクトは含まれません。

- 演算子
- 役割

グローバル・コンテンツとリンク項目

このリリースでは、グローバル・オブジェクト・マッピングは、外部コンテンツ・サイト、Fixlet (およびタスク) および分析に限定されます。カスタム・コンテンツのリンクは、このリリースの対象外です。

Insights データベースでは、1 つ以上の BigFix Enterprise データベース・データ・ソースからデータをインポートできます。このリリースでは、最大 10 個のデータ・ソースからのデータの取り込みがサポートされており、取り込まれるデータ・ソースのエンドポイントの総数が 250,000 を超えないようにする必要があります。

グローバル・オブジェクトの概念では、複数の BigFix Enterprise データベース・データ・ソースからの同じオブジェクトは 1 つであると見なされます。例えば、BigFix Enterprise サーバー・サポート・サイトは、1 つのデータ・ソースでも、複数のデータ・ソースでも、どのデータ・ソースでも同一と見なされます。

サイジングの概念

Insights のスケーラビリティとサイズの要件は、データ・セットの概念に直接関連しており、「変化率」の要素が追加されています。

それぞれの環境は固有であるため、これらのデータ・セットのサイズは環境ごとに異なります。例えば、プロパティーの結果データの量が多い環境の変化率は、Fixlet データのサイズが大きい環境とは異なる方法でデータベースに影響を与えます (プロパティーの結果は戻されるデータの点で異なりますが、Fixlet データはバイナリーです)。データの幅の変化は、これら 2 つのデータ・タイプの間で大きく、サイズにも影響します。サイズ変更に関する考慮事項は、常に多くの要因に依存します。データ・ソースには動的な性質があるため、取り込まれてリンクされるオブジェクトの数によって最初の ETL フ

フル・データ・セットが決定され、変化率によってその後の変更が決定されます。したがって、サイズ変更のアプローチは、主に最初のフル・データ・セットのサイズに焦点を合わせています。変更のセットは変更率(パーセント)として受け入れられ、前提としてフル・データ・セットに適用されます。この前提により、その後の ETL の変更に伴うデータベースの成長率が予測されます。概念的には、次のカテゴリーを考慮してください。

名前	説明
初期レポート・データセット	初期レポート・データセットを提供します。このデータセットは、時間の経過に伴ってデータベースのサイズに最も影響を与えます。
ETL (初期)	履歴データまたは特定の時点のデータセットを提供します。このデータセットは、最後の ETL 以降に変更された値のみ取得します。この ETL タイプは、ソース・データ・ソースの ETL (特定の時間点および変更) 变化率に左右されます。変化率が最小である場合、ETL の処理に必要な時間に加えて、拡張定の影響も最小限に抑えられます。しかし、変化率が大きい場合は、ストレージ・リソースと ETL 完了時間が影響を受けます。

一般的な傾向として、初期データ・ソースのサイズが、結果として得られる Insights データベースのサイズ(最初の ETL の後)に関連付けられます。この結果のサイズは異なる場合があります。ただし、次のクエリーは、Insights が必要とするデータベース・コンテナーのサイズを大まかに見積もります。取り込み中の BigFix Enterprise データベースに関する添付されたクエリーを実行します。返される数値は、初期データセットが 20% 増加した後で控えめに見積もられた値です。すべての取り込み中の BigFix Enterprise 環境でクエリーを実行して、最初の ETL の累積サイズを予測します。

Query-FullRun の見積もり

```
-- SCRIPT USED to Guestimate 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
```

スケールの影響、ベスト・プラクティス、および最適化

時間の経過に伴うデータの影響を最小限に抑え、ETL 处理時間を最適化するために、次の原則を考慮すると、良好な結果が得られます。

- 取り込まれたサイトの量を減らします (データ・セットごとに提供されているダイアログ)。データの履歴をレポートにしない場合は、データを Insights にインポートしないようにします。
- プライマリー・サイトを追加すると、そのデータは履歴的に保持され、削除できません。BigFix Insights の初期セットアップ時にインポートするサイトを計画します。
- ビジネス・オペレーションに応じた頻度で ETL を実行する計画を立てます。推奨される頻度は 1 日 1 回です。
- 評価する予定のコンピューターへのチェックリスト・サイトのサブスクリプションを減らします。

システム・バージョン管理テーブル

BigFix Insights が作成するデータ・ウェアハウスは、受信データと変更データの履歴を記録するために SQL Server のシステム・バージョン管理テーブルを活用します。

Insights によって作成されるすべての一時テーブルのデフォルトの保持ポリシーは 36 ヶ月です。現時点では、これを修正する唯一の方法は手作業で行うことです。ただし、WebUI 設定「_WebUIAppEnv_RETENTION_PERIOD」を他のサポートされている値 (100 日など) に適用することは可能です。Insights データ・レイクを生成する前にこの設定を適用すると、すべての一時テーブルはデフォルトではなくこの新しい値を使用します。事後的に適用した設定は、それ以降に生成されるテーブルにのみ影響します。

WebUI Insights V12 の時点で、以下のテーブルには既知の問題があります。

```
[dbo].[device_dimensions_historical]
```

保持ポリシーが無限に設定されています。この問題は今後のリリースで修正される予定です。

SQL トランザクションとアプリケーションの動作パターン

すべてのトランザクションは、インポート時にスナップショット分離を使用します。その時点から、SQL は準備されたバッチ・ステートメントを使用し、このステートメントは upsert に取り込まれます。upsert は、次の 2 つの機能を実行します。1 つ目は、適用可能な宛先の行を読み取って更新が必要かどうかを確認することで、2 つ目は必要に応じて行を挿入または更新することです。インポート中に、tempdb が頻繁に使用されます。データは tempdb にダンプされ、その後、読み取られて、更新または挿入のために処理されます。

インポート・プロセスの最終段階で、グローバル・オブジェクトが計算され、保存される（ウェアハウス）ときに、読み取りの影響は大きく（約 90 %）、書き込み影響は小さくなります（約 10 %）。

通常のビジネス・インテリジェンス・レポートの実行アクティビティーでは、読み取りの影響が大きくなり、書き込み操作はほとんど発生しません。

前述の情報を考慮すると、Insights DBMS のバランスは、アプリケーションの使用状況に応じて読み取りをやや優先し、75% ~ 25% の範囲で最適化することが推奨されます。このため、以下の仕様が定義されています。

BigFix Insights SQL 構成のガイドライン

構成	説明または根拠
tempdb I/O チャネルを共有ではなく、専用にします。	Insights は tempdb を頻繁に活用し、可能な場合は他のワークロードから TempDB を分離します。
SQL メモリーの制限を正しく設定します。	オペレーティング・システムに 8 GB 以上を許可するように SQL メモリーが制限されていることを確認します。この上限は、SQL Server のプロパティーで設定されます。
すべてのデータ・ファイル・セットを含む、SQL ファイルの格納場所を除外するようにウィルス・スキャナーを構成します。	アンチウィルス・ソフトウェアの設定を構成する場合は、以下のファイルまたはディレクトリー（該当する場合）をウィルス・スキャンから除外します。除外すると、ファイルのパフォーマンスが向上し、SQL Server サービスでファイルを使用する必要がある場合にファイルがロックされないようにします。この除外ルールの設定方法について詳しくは、ウィルス・スキャナーの説明のリンクを参照してください。アンチウィルス・ソフトウェアの選択については、「 SQL Server を実行しているコンピューターで使用するアンチウィルス・ソフトウェアを選択する方法 」を参照してください。
サポートする SQL データ・ファイルに対しても、ファイルのインデックス作成やファイル圧縮を使用しないでください。	アンチウィルス・ファイルの除外の理由と同様の理由を、HIPS（ホスト侵入防止システム）ベースのアプリケーションや、使用中のデータ・ファイルをロックする可能性のあるファイル・インデックス処理に適用します。

最初に、トランザクション・ログと MDF ファイルのサイズをサ

イズ予測の 80%
に設定し、自動拡張を適切に設定します。

SQL データベース・ファイルの自動拡張を、最小数ではなく、実質的な数に設定します。サポートする DB ファイルは、Insights データベースをサポートする SQL 内では非常にサイズが大きくなる可能性があります。拡張するサポート対象のデータ・ファイルに対してシステムが使用する時間とリソースを最小限に抑えるため、拡張特性が自動拡張のデフォルト設定から、2 GB 以上または毎回 10 % ずつに変更されるようになります。ほとんどの場合、パーセンテージによるアプローチは最良の結果をもたらします。

ソフト NUMA が準備されていることを確認します。Microsoft によると、SQL 2017 と 2019 はソフト NUMA を使用せず、デフォルトで NUMA 構成を管理します。この設定がデフォルト設定から変更されていることを確認します。詳細については、「[ソフト NUMA \(SQL Server\)](#)」を参照してください。

ETL のリズムとスケジューリング

ETL スケジュールは、Insight 管理者を通じて制御できます。ETL の実行頻度が低いほど、変更の割合が増加するため、ETL の実行期間が長くなります。逆に、ETL の実行頻度が高いほど、ETL の実行期間が短くなる可能性が高くなります。ただし、実行頻度が高いほど、記録される傾向データが増えるため、消費される領域が増える可能性があります。期間や頻度と、サイズやストレージの要件の相関関係は、必ずしも 1 つの式で予測できるものではなく、BigFix Enterprise Server 環境、ビジネス・プラクティス、条件に基づいて大きく異なります。最良の結果を得るには、最初の ETL を実行した後、(フィルター処理されたサイトに基づいて) 同じ定数データ・セットで ETL を毎日実行します。次に、データベースのサイズの増分と ETL の期間を毎日測定します。これはインポート画面で確認できます。この作業は、時間の経過に応じた変化の割合を定義するのに役立ちます。このメトリックを使用すると、計算された ETL の頻度に基づいて予測されるデータベースのサイズを決定できます。次に、データがキャプチャされ、記録されます。ストレージ要件と ETL の頻度のバランスをさらに理解し、ETL の期間をより適切に計算できます。

**Note:**

ETL は同時にスケジュールできますが、初期リリースではデータ・ソースから ETL をキューに入れます。ETL スケジュールを定義する場合は、このシナリオを検討してください。

複製を管理する

BigFix Insights は、Insights データベースと同じ場所に配置されている BigFix Enterprise データ・ソースの複製を使用します。Insights の WebUI コンポーネントは、指定された WebUI サーバー上に配置されています。これは取り込まれた BigFix Enterprise Server の 1 つであると考えられます。BigFix Insights データベースは、同じ WebUI サーバー上に同時に配置することもできます。WebUI サーバーにデータベースを配置することを技術的に妨げるものはありません。WebUI サーバーは BigFix Insights データベースへの通信プロセスを維持し、取り込まれた情報と監査の詳細はすべて Insights データベースに格納されます。Insights データベースは複数の BigFix Enterprise データベースを消費する可能性があります。また、WebUI サーバーと同じ場所に配置されている場合は、Insights データベース上の他の BigFix Enterprise データ・ソースに情報境界を拡張する可能性があります。

バックアップ・アプローチ

潜在的なダウンタイムを短縮し、リカバリー間隔を改善するには、既存の BigFix Enterprise データベースのバックアップが必要です。自動化を使用して、BigFix Enterprise データベースを 1 次 BigFix サーバーから定期的にバックアップし、別の SQL Server 上のバックアップ・セットを、時間とレプリカのスナップショットとしてリストアすることができます。使用可能なツールとユーティリティーに応じて、スナップショットとレプリカをさまざまな方法で生成できます。

次のリストでは、共通のツール・セットを使用する 1 つのバックアップ方法について説明します。

- BigFix サーバー: 定期的なデータベース・バックアップ (BigFix Enterprise と BigFix Enterprise サーバー・レポート)。
- 1 次 BigFix サーバー: 定期的なスケジュールされたタスク。スタンバイ・サーバーへの Robocopy データベース・バックアップと BigFix Enterprise サーバーのバックアップを生成します。
- BigFix レプリカ・サーバー: 定期的なデータベース・リストア (BigFix Enterprise および BigFix Enterprise サーバー・レポート)。

BigFix サーバー - 定期データベース・バックアップ (**BFEEnterprise**)

**Note:**

次のサンプル・スクリプトをご使用の環境で実行するには、編集が必要になる場合があります。

```

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
```

1 次 BigFix サーバー - 定期的なスケジュールされたタスク: スタンバイ・サーバーへの Robocopy データベースと BES サーバーのバックアップ



Note:

これらのコマンドは、コマンド・プロンプトまたは PowerShell 環境から実行されます。

```
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 レプリカ・サーバー - 定期データベース・リストア (BFEEnterprise)



Note:

次のサンプル・スクリプトをご使用の環境で実行するには、編集が必要になる場合があります。

```

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
```

SQL と SAN の追加

SAN に関する考慮事項

SAN ベースの環境に SQL Server をインストールして構成する場合は、いくつかの要素を考慮する必要があります。SQL と SAN の両方について理解している必要があります。

SAN をインストールする際には、SQL Server の基本事項を理解している必要があります。SAN が使用されているかどうかにかかわらず、SQL ファイルが格納されているハードウェアについて理解していることが重要です。構成値が基本値と矛盾しないようにする必要があります。

- SAN ハードウェアはすべて、SQL Server I/O の基本に準拠している必要があります。書き込み順序を維持する必要があります。つまり、書き込みは常に安全な場所または安定したメディアで保護される必要があります。また、I/O の書き込み動作が中断されないように予防措置を講じる必要があります。書き込み順序のプロセスは、選択した製造元または製造元の SAN ドキュメントから入手できる必要があります。

1. 安定したメディアとは、システムの再起動や一般的な障害に耐えられるストレージのことです、物理的なディスク・ストレージが該当しますが、SAN キャッシュもこの定義に含まれます。安定したメディアは SQL Server の要件であり、サーバーはデータベースの ACID プロパティを維持し、データの整合性を確保するために、ログ先行書き込み用の安定したメディアを使用します。安定したメディア戦略は、ハードウェアの製造元によってサポートされる必要があります。
2. 書き込み順序は、書き込み操作の順序を保持する機能であり、ハードウェア上の SQL Server の要件です。SQL Server では、ローカルおよびリモートの I/O ターゲットに対して書き込み順序を維持する必要があります。書き込み順序が維持されない場合、システムはログ先行書き込みプロトコルを中断します。
3. 中断された I/O とは、部分的な書き込みが行われ、データが無効な状態のままになる状況を指します。SQL Server では、このページは「中断されたページ」と説明しています。中断されたページまたは I/O を防止するとは、I/O パス内のさまざまな I/O エンティティーに I/O を分割するなど、中断された I/O を防ぐために、ディスク・システムがセクターの配置やサイズ設定を行う必要があることを意味します。

SAN および RAID 構成

SAN ハードウェアはゾーニングをサポートしており、管理者はこれを使用して、I/O が SQL Server から分離されるように、完全な物理ディスクを LUN 専用にすることができます。SQL Server のパフォーマンスは、物理ディスクへの読み取り/書き込みアクセスに依存します。したがって、SAN にサーバーをデプロイする場合は、基本的な RAID の基礎が重要です。SQL は I/O を大量に使用するアプリケーションであり、パフォーマンスを最優先すべきなのは SQL Server のディスク構成です。Microsoft では、SQL ログ・ファイルを物理ディスク・レベルのデータ・ファイルから分離することを推奨しています。SAN 環

境では、これはデータベース・ログ・ファイルを、他の LUN(他のデータベース・ログ、データ、インデックス・ファイルに使用されている LUN を含む)と共有されていない LUN から、専用の物理ディスクを持つ LUN に分離することを意味します。Microsoft のベスト・プラクティスによると、この構成では、書き込みの応答時間を 2~3 ミリ秒、読み込みの応答時間を 20 ミリ秒以下にすることが目標とされています。サーバーやアプリケーションが共有する物理ディスクを分離します。これは、I/O 特性の異なるアプリケーションを実行する異なるサーバーが問題を引き起こす可能性があるためです。

その他の考慮事項

SANS を使用する大規模なデータベース・サーバーでは、TempDB、ログ、およびデータベース・ファイル・サブシステムを物理的なディスク・スペースの要件から分離します。大規模なデータベース・エンジンは、通常 TempDB やトランザクション・ログを使用するため、SQL の TempDB やトランザクション・ログ・ファイルを、専用の RAID サブシステム(できれば、RAID10)に分離することを追加で考慮する必要があります。Microsoft SQL のベスト・プラクティスでは、MDF とインデックス・ファイルには RAID 1+0 が推奨されています。また、RAID5 は推奨されていませんが、これは RAID5 に関連する読み取り/書き込み時間が長くなるためです。Microsoft の推奨事項に基づいて、250 Kb のストライプ・サイズを使用してください。ストライプ・サイズは単一の SQL エクステントのサイズであるため、64 Kb より小さく設定しないでください。

ストレージ要件

特定のコンポーネントに対して別個のストレージ・サブシステムとコントローラーを使用することには利点があります。最良の結果を得るために、ベース、データベース・ログ、データベース・コンテナーのそれぞれに個別のストレージ・サブシステムを設定してください。データベース・ログは、非常に高速な順次 I/O を必要とするという特徴があります。データベース・コンテナーでは、より多様なアクセス・パターンを使用できますが、データベース・バッファー・プールがあるため、ストレージへの影響はほとんどありません。

既知の問題と制約

Insights データ・レイク

- 現在、データソース認証は SQL 認証に制限されます。
- リンク・オブジェクトは、外部サイト、Fixlet、タスク、分析に制限されます。



Note:

カスタム・サイトはサポートされていません。

- BigFix Enterprise サーバーのオペレーター、役割、および非管理資産は、Insights によって取り込まれません。
- Insights は、パフォーマンスの条件に応じて、8つ以下の BigFix Enterprise データベースからのデータの取り込みをサポートしています。

Tableau サンプル・レポート

- 大きいデバイス・セットのデータを表示する Tableau レポートのパフォーマンスにはらつきがあります。
- Tableau サンプル・レポートは、ETL が正常に完了した BigFix Insights データベースでのみ使用できます。レポートは、ETL が毎日スケジュールされているかのように実行されます。

詳細情報

詳細については、次のドキュメント・セットを参照してください。

- [BigFix Insights](#)
- [BigFix フォーラム](#)

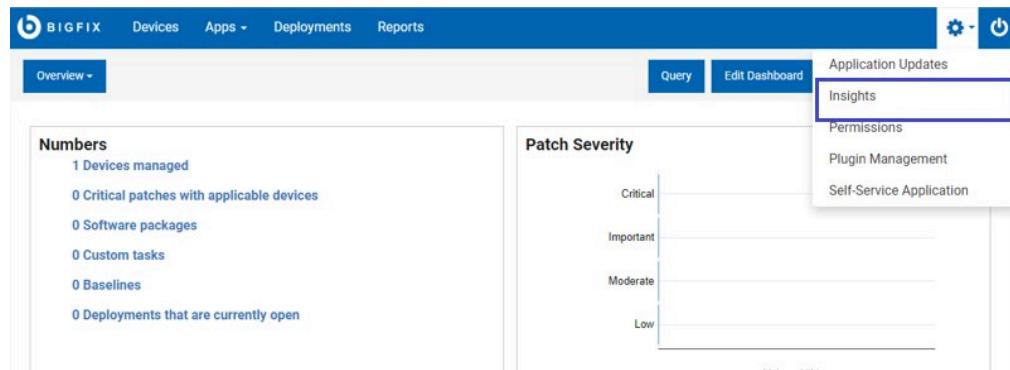
Chapter 4. BigFix Insights のセットアップ

BigFix 10 Insights をセットアップし、BigFix 10 Insights の接続の詳細を編集する方法について説明します。

BigFix Insights の初期設定は、WebUI アプリ内で行われます。Insights をインストールして構成するには、デプロイメント内のマスター・オペレーターとして WebUI Insights アプリケーションに移動します。

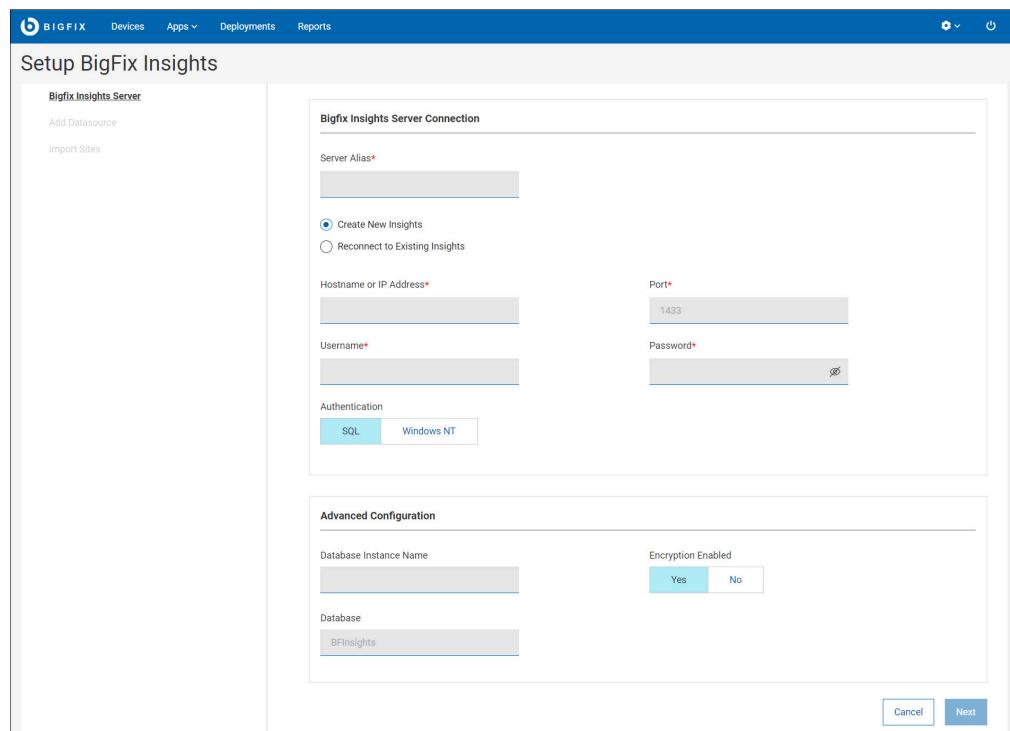
1. マスター・オペレーター資格情報を使用して **WebUI** にログインします。
2. ナビゲーション・バーの歯車アイコンをクリックします。

Figure 2. Insights へのナビゲート



3. 「Insights」をクリックします。
「BigFix Insights の設定」ページが表示されます。

Figure 3. BigFix Insights データベースへの接続

**Note:**

「新規 Insights の作成」オプションが選択されていることを確認します。

4. Insights Datalake の場所を定義するために、次の接続プロパティーを指定します。

サーバーのエイリアス

Insight のアプリケーションの名前。

**Note:**

別名は、データベース名またはサーバー名に対応していません。これは、WebUI アプリ内でこの Insights Datalake インスタンスを参照するための共通名を指定するために使用されます。別名の名前は変更できません

ホスト名または IP アドレス

Insights Datalake を割り当てるターゲット・データベース・サーバー。

ポート

Insights Datalake をホストする SQL Server のリスニング・ポート (デフォルトでは、SQL は 1433 で listen します)。

ユーザー名

接続先のターゲット・データベース・サーバー上に Insights Datalake を作成する権限を持つユーザー名。

パスワード

ユーザー名に関連付けられているパスワード。

認証

Insights Datalake を認証および作成するための認証タイプ。

詳細な構成

パラメーター (必要な場合)。

データベースのインスタンス名

代替 SQL またはサーバー・インスタンス名。



Note:

デフォルトでは、SQL はデフォルトのインスタンスを利用します。この構成は、SQL Server のインストール時に定義されます。このフィールドに値が入力されていない場合 (デフォルトの動作)、デフォルトの MSSQL インスタンスは Insights Datalake を作成するターゲットとなります。

接続の暗号化

接続を暗号化するか、暗号化しないかを定義します。



Note:

接続は暗号化され、これがデフォルトの選択です。

データベース名

Insights Datalake のデフォルト名は **BFInsights** です。このデフォルトのデータベース名の値からデフォルトの Insights Datalake 名を変更する場合は、優先される名前を定義する必要があります。



Note:

データベース管理者は、対応するデータベース・ファイル・セットに関連する論理ボリュームを指定することを選択します。これは Microsoft のベスト・プラクティスです。例: トランザクション・ログとデータベース・コン



テナー・ファイルは、論理ボリュームと物理ボリュームに分離して配置する必要があります。データベース管理者が Insights のインストールを開始する前にデータベースを作成し、(このフィールドに表示される値と)同じ指定された値でデータベースに名前が付けられている場合、インストールによって目的のデータベースに Insights スキーマが入力され、その結果、DBA が選択したファイルの場所が継承されます。これは、Insights Datalake の新規インストールにのみ適用されます。詳しくは、DBA に相談してください。

5. 「次へ」をクリックします。

これにより、定義された接続の詳細を使用して、定義された Insights Datalake がサーバー上に作成されます。この手順を完了すると、初期データベースが作成されます。

この Insights Datalake のプロビジョニングは完了します。Insights サーバーのホーム・ページが表示され、ここでユーザーは最初の **BFEEnterprise** ソースを Insights に追加することができます。Insights には、ターゲットの BFE レプリカの詳細と、最初のデータ・ソースを取り込むための資格情報が必要です。データ・ソースの構成については、「データ・ソースとサイトの操作」を参照してください。

BigFix Insights 接続の詳細の編集

Insights Datalake の場所または接続情報を変更したり、Insights の資格情報を変更したり（パスワードの変更を含む）する必要がある場合は、Insights のランディング・ページの「設定」の「接続の編集」ボタンを使用します。このページでは、現在の接続情報を表示したり、接続情報を変更したりできます。Insights Datalake の接続の詳細を表示または編集するには、以下のステップを実行します。

1. Insights サーバーのホーム・ページで、「設定」タブをクリックして接続の詳細を表示します。

The screenshot shows the 'Connections' section of the Insights Settings page. The 'Datasource' tab is selected. A table displays connection information for 'tvitinsight':

BigFix Insights Database Connection Information	
Alias	tvitinsight
Database Name	BFinights
Domain	tvitinsight
Authentication	Windows NT
Hostname/IP	tvitinsight
SQL Server Instance Name	N/A
Port	1433
Username	administrator
Connection Encrypted	Yes

A blue 'Edit Connection' button is located at the bottom left of the table.

2. 接続の詳細や認証に使用される資格情報を変更または編集するには、「接続の編集」をクリックします。

**Note:**

接続の詳細を編集する前に、このページで使用できる機能を理解しておく必要があります。このページは、Insights DB が配置されている構成パラメーターを制御するため、接続の詳細が誤っていると悪影響を及ぼす可能性があります。フィールドとその目的については、[BigFix Insights のセットアップ](#) のステップ 4 で説明されています。

Edit BigFix Insights connection

Bigfix Insights Server Connection

Server Alias*	tvitinsight
Domain*	tvitinsight
Hostname or IP Address*	tvitinsight
Port*	1433
Username*	<input type="password"/>
Authentication	
SQL	Windows NT

Advanced Configuration

Database Instance Name	Encryption Enabled
<input type="text"/>	<input checked="" type="radio"/> Yes <input type="radio"/> No
Database	<input type="text"/> BFInsights

Buttons: Cancel Save

3. 値を目的の構成に編集したら、「保存」をクリックします。または、「キャンセル」をクリックして終了します。

Chapter 5. BigFix Insights のアンインストール

BigFix Insights をアンインストールする方法については、以下の手順を参照してください。

1. データベース・クリーンアップ

以下の行を dbo.webuidata テーブルから削除します

```
delete from [BFEEnterprise].[dbo].[WEBUI_DATA] where App='insights'  
delete from [BFEEnterprise].[dbo].[WEBUI_DATA] where App='ivr' and Name = 'dataProgression'
```

2. IVR キャッシュ・テーブルのクリア

以下の IVR キャッシュ・テーブルをクリアします。

```
delete from webui.IVR_VULNERABILITY_META  
delete from webui.IVR_VULNERABILITY_FIXLET_MAP  
delete from webui.IVR_CVE_FIXLET_MAP
```

3. WebUI Insights の再起動

WebUI Insights を再起動します。これを実行するには、WebUI を再起動するか、Windows タスク・マネージャーから WebUI Insights ノード・プロセスを終了します。

4. (オプション): Insights データベースの削除

この手順は、Insights データベースのデータが不要になり、削除しても問題ないことが確実な場合にのみ実行してください。