Red Hat Data Grid 8.3

Data Grid のアップグレード

Data Grid を 8.3 にアップグレード

Last Updated: 2023-12-06

Red Hat Data Grid 8.3 Data Grid のアップグレード

Data Grid を 8.3 にアップグレード

法律上の通知

概要

Data Grid クラスターをある 8.x バージョンから別のバージョンにアップグレードします。ローリングアップグレードを実行して、Data Grid が互換性のためにデータを変換するダウンタイムまたはオフラインアップグレードを回避することができます。

目次

RED HAT DATA GRID

Data Grid は、高性能の分散型インメモリーデータストアです。

スキーマレスデータ構造

さまざまなオブジェクトをキーと値のペアとして格納する柔軟性があります。

グリッドベースのデータストレージ

クラスター間でデータを分散および複製するように設計されています。

エラスティックスケーリング

サービスを中断することなく、ノードの数を動的に調整して要件を満たします。

データの相互運用性

さまざまなエンドポイントからグリッド内のデータを保存、取得、およびクエリーします。

DATA GRID のドキュメント

Data Grid のドキュメントは、Red Hat カスタマーポータルで入手できます。

- Data Grid 8.3 ドキュメント
- Data Grid 8.3 コンポーネントの詳細
- Data Grid 8.3 でサポートされる設定
- Data Grid 8 機能のサポート
- Data Grid で非推奨の機能

DATA GRID のダウンロード

Red Hat カスタマーポータルで Data Grid Software Downloads にアクセスします。

注記

Data Grid ソフトウェアにアクセスしてダウンロードするには、Red Hat アカウントが必要です。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ を参照してください。

第1章 DATA GRID 8 のアップグレードに関する注意点

本セクションの詳細を確認してから 1 つの Data Grid 8 バージョンから別のバージョンにアップグレードしてください。

1.1. DATA GRID 8.3 へのアップグレード

以下の情報を参照して、以前のバージョンの Data Grid 8 から 8.3 へのアップグレードが成功するようにします。

ファイルベースのキャッシュストアを使用したデプロイメントのアップグレード

Data Grid 8.3 以降、ファイルベースのキャッシュは、シングルファイルである **SingleFileStore** ではなく、ソフトインデックスである **SoftIndexFileStore** に保存します。Data Grid 8.2 以前では、**SingleFileStore** はファイルベースのキャッシュストアのデフォルトでした。

スキーマバージョンの **13.0** 以降を持つ **file-store** 設定は、アップグレード後に **SoftIndexFileStore** に自動的に移行されます。

以前のバージョンから Data Grid 8.3 にアップグレードし、キャッシュに **soft-index-file-store** 要素の設定が含まれる場合は、その設定を代わりに **file-store** 要素を使用するように変換する必要があります。スキーマの変更および Data Grid キャッシュ設定の変更に関する詳細は、移行に関するドキュメントを参照してください。

関連情報

- Data Grid 8 への移行

複数のエンドポイント設定のある Data Grid Server デプロイメントのアップグレード

Data Grid Server 8.3 以降では、セキュリティーレルムを使用してエンドポイントを設定し、**endpoint** 要素を使用して Hot Rod または REST コネクターを設定します。**endpoints** 要素は、複数の **endpoint** 要素をラップします。

8.2 以前から Data Grid Server をアップグレードし、設定に **endpoints** 要素の子要素が含まれている場合は、**endpoint** 要素を使用するように設定を移行する必要があります。

関連情報

- Data Grid 8 への移行

最低でも 8.1 からのアップグレード

8.0 からアップグレードする場合は、最初に 8.1 にアップグレードする必要があります。Data Grid 8.0 の永続データは、新しいバージョンのバイナリーではありません。この非互換性の問題に対処するために、Data Grid 8.2 以降では、クラスター起動時に Data Grid 8.1 から既存の永続キャッシュストアを自動的に変換します。ただし、Data Grid は、Data Grid 8.0 からキャッシュストアを変換しません。

第2章 DATA GRID SERVER クラスターのローリングアップグレードの実行

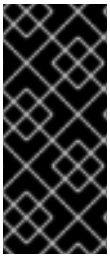
Data Grid クラスターのローリングアップグレードを実行して、ダウンタイムやデータの損失なしにバージョン間で変更し、Hot Rod プロトコルを介してデータを移行します。

2.1. ターゲット DATA GRID クラスターの設定

アップグレードする予定の Data Grid バージョンを使用するクラスターを作成してから、リモートキャッシュストアを使用してソースクラスターをターゲットクラスターに接続します。

前提条件

- ターゲットクラスターに必要なバージョンの Data Grid Server ノードをインストールします。



重要

ターゲットクラスターのネットワークプロパティーはソースクラスターのネットワークプロパティーが重複していないことを確認します。JGroups トランスポート設定でターゲットおよびソースクラスターの一意の名前を指定する必要があります。環境に応じて、異なるネットワークインターフェイスとポートオフセットを使用して、ターゲットクラスターとソースクラスターを分離することもできます。

手順

1. ターゲットクラスターがソースクラスターに接続できるリモートキャッシュストア設定を JSON 形式で作成します。
   ターゲットクラスターのリモートキャッシュストアは、Hot Rod プロトコルを使用して、ソースクラスターからデータを取得します。

```
{
  "remote-store": {
    "cache": "myCache",
    "shared": true,
    "raw-values": true,
    "security": {
      "authentication": {
        "digest": {
          "username": "username",
          "password": "changeme",
          "realm": "default"
        }
      }
    },
    "remote-server": [
      {
        "host": "127.0.0.1",
        "port": 12222
      }
    ]
  }
}
```

2. Data Grid コマンドラインインターフェイス (CLI) または REST API を使用して、リモートキャッシュストア設定をターゲットクラスターに追加し、ソースクラスターに接続できるようにします。

   - CLI: ターゲットクラスターで **migrate cluster connect** コマンドを使用します。

     ```
     [//containers/default]> migrate cluster connect -c myCache --file=remote-store.json
     ```

   - REST API: **rolling-upgrade/source-connection** メソッドを使用して、ペイロードにリモートストア設定が含まれる POST リクエストを呼び出します。

     ```
     POST /v2/caches/myCache/rolling-upgrade/source-connection
     ```

3. 移行するキャッシュごとに直前の手順を繰り返します。

4. すべての要求の処理を開始するために、クライアントをターゲットクラスターに切り替えます。

   a. クライアント設定をターゲットクラスターの場所で更新します。

   b. クライアントを再起動します。

関連情報

- リモートキャッシュストア設定スキーマ

2.2. ターゲットクラスターへのデータの同期

ターゲット Data Grid クラスターを設定してソースクラスターに接続する場合、ターゲットクラスターはリモートキャッシュストアを使用してクライアント要求を処理し、オンデマンドでデータをロードできます。データをターゲットクラスターに完全に移行して、ソースクラスターの使用を停止できるようにするには、データを同期します。この操作はソースクラスターからデータを読み取り、ターゲットクラスターに書き込みます。データは、ターゲットクラスターのすべてのノードに並行して移行され、各ノードはデータのサブセットを受け取ります。ターゲットクラスターに移行する各キャッシュの同期を実行する必要があります。

前提条件

- 適切な Data Grid バージョンでターゲットクラスターを設定します。

手順

1. ターゲットクラスターに移行する各キャッシュと Data Grid コマンドラインインターフェイス (CLI) または REST API との同期を開始します。

   - CLI: **migrate cluster synchronize** コマンドを使用します。

     ```
     migrate cluster synchronize -c myCache
     ```

   - REST API: POST リクエストと共に **?action=sync-data** パラメーターを使用します。

     ```
     POST /v2/caches/myCache?action=sync-data
     ```

操作が完了すると、Data Grid はターゲットクラスターにコピーされたエントリーの合計数で応答します。

2. ターゲットクラスター内の各ノードをソースクラスターから切断します。

- CLI: **migrate cluster disconnect** コマンドを使用します。

```
migrate cluster disconnect -c myCache
```

- REST API: DELETE リクエストを呼び出します。

```
DELETE /v2/caches/myCache/rolling-upgrade/source-connection
```

次のステップ

ソースクラスターからすべてのデータを同期した後、ローリングアップグレードプロセスが完了しました。ソースクラスターの使用を停止できるようになりました。

第3章 キャッシュストア間のデータの移行

Data Grid は、キャッシュストア間で永続化されたデータを移行するための Java ユーティリティーを提供します。

Data Grid をアップグレードする場合、メジャーバージョン間の機能相違点は、キャッシュストア間の後方互換性を許可しません。**StoreMigrator** を使用してデータを変換し、ターゲットバージョンとの互換性を持つことができます。

たとえば、Data Grid 8.0 にアップグレードすると、デフォルトのマーシャラーが Protostream に変更されます。以前の Data Grid バージョンでは、キャッシュストアはバイナリー形式を使用し、マーシャリングする変更との互換性がありません。つまり、Data Grid 8.0 は、以前の Data Grid バージョンでキャッシュストアから読み込むことができません。

他の場合は、Data Grid のバージョンが、JDBC Mixed および Binary ストアなどのキャッシュストア実装を非推奨または削除します。このような場合は、**StoreMigrator** を使用して異なるキャッシュストア実装に変換できます。

3.1. キャッシュストアマイグレーター

Data Grid は、最新の Data Grid キャッシュストア実装のデータを再作成する **StoreMigrator.java** ユーティリティーを提供します。

StoreMigrator は以前のバージョンの Data Grid のキャッシュストアを取得し、キャッシュストア実装をターゲットとして使用します。

StoreMigrator を実行すると、**EmbeddedCacheManager** インターフェイスを使用して定義したキャッシュストアタイプでターゲットキャッシュが作成されます。**StoreMigrator** は、ソースストアからメモリーにエントリーを読み込み、それらをターゲットキャッシュに配置します。

StoreMigrator を使用すると、あるタイプのキャッシュストアから別のストアにデータを移行することもできます。たとえば、JDBC String ベースのキャッシュストアから RocksDB キャッシュストアに移行することができます。

重要

StoreMigrator は、セグメント化されたキャッシュストアから以下にデータを移行できません。

- 非セグメント化されたキャッシュストア。
- セグメント数が異なるセグメント化されたキャッシュストア。

3.2. キャッシュストアマイグレーターの取得

StoreMigrator は、Data Grid ツールライブラリー **infinispan-tools** の一部として利用でき、Maven リポジトリーに含まれます。

手順

- **StoreMigrator** の **pom.xml** を以下のように設定します。

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
```

```
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.infinispan.example</groupId>
  <artifactId>jdbc-migrator-example</artifactId>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>org.infinispan</groupId>
      <artifactId>infinispan-tools</artifactId>
    </dependency>
    <!-- Additional dependencies -->
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>1.2.1</version>
        <executions>
          <execution>
            <goals>
              <goal>java</goal>
            </goals>
          </execution>
        </executions>
        <configuration>
          <mainClass>org.infinispan.tools.store.migrator.StoreMigrator</mainClass>
          <arguments>
            <argument>path/to/migrator.properties</argument>
          </arguments>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

3.3. キャッシュストアマイグレーターの設定

ソースおよびターゲットのキャッシュストアのプロパティーを **migrator.properties** ファイルに設定します。

手順

1. **migrator.properties** ファイルを作成します。
2. ソースキャッシュストアを **migrator.properties** に設定します。
   a. 以下の例にあるように、すべての設定プロパティーの先頭に **source.** を追加します。

```
source.type=SOFT_INDEX_FILE_STORE
```

```
source.cache_name=myCache
source.location=/path/to/source/sifs
source.version=<version>
```

3. **migrator.properties** でターゲットキャッシュストアを設定します。

   a. 以下の例のように、すべての設定プロパティーの先頭に **target.** を付けます。

```
target.type=SINGLE_FILE_STORE
target.cache_name=myCache
target.location=/path/to/target/sfs.dat
```

3.3.1. キャッシュストアマイグレーターの設定プロパティー

ソースおよびターゲットのキャッシュストアを **StoreMigrator** プロパティーで設定します。

表3.1 キャッシュストアタイププロパティー

プロパティー	説明	必須/オプション
type	ソースまたはターゲットのキャッシュストアタイプのタイプを指定します。 **.type=JDBC_STRING** **.type=JDBC_BINARY** **.type=JDBC_MIXED** **.type=LEVELDB** **.type=ROCKSDB** **.type=SINGLE_FILE_STORE** **.type=SOFT_INDEX_FILE_STORE** **.type=JDBC_MIXED**	必須

表3.2 一般的なプロパティー

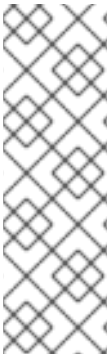
プロパティー	説明	値の例	必須/オプション
cache_name	ストアがバックアップするキャッシュに名前を付けます。	**.cache_name=myCache**	必須

プロパティー	説明	値の例	必須/オプション
segment_count	セグメンテーションを使用できるターゲットキャッシュストアのセグメント数を指定します。 セグメント数は、Data Grid 設定の **clustering.hash.numSegments** と一致する必要があります。 つまり、キャッシュストアのセグメント数は、対応するキャッシュのセグメント数と一致する必要があります。セグメントの数が同一でない場合、Data Grid はキャッシュストアからデータを読み込めません。	**.segment_count=256**	Optional

表3.3 JDBC プロパティー

プロパティー	説明	必須/オプション
dialect	基礎となるデータベースのダイアレクトを指定します。	必須
version	ソースキャッシュストアのマーシャラーバージョンを指定します。 以下のいずれかの値を設定します。 * Data Grid 7.2.x の場合は **8** * Data Grid 7.3.x の場合は **9** * Data Grid 8.0.x の場合は **10** * Data Grid 8.1.x の場合は **11** * Data Grid 8.2.x の場合は **12** * Data Grid 8.3.x の場合は **13**	ソースストアにのみ必要です。
marshaller.class	カスタムマーシャラークラスを指定します。	カスタムマーシャラーを使用する場合に必要です。

プロパティー	説明	必須/オプション
marshaller.externalizers	**[id]:<Externalizer class>** 形式で読み込むカスタム **AdvancedExternalizer** 実装のコンマ区切りリストを指定します。	Optional
connection_pool.connection_url	JDBC 接続 URL を指定します。	必須
connection_pool.driver_class	JDBC ドライバーのクラスを指定します。	必須
connection_pool.username	データベースユーザー名を指定します。	必須
connection_pool.password	データベースユーザー名のパスワードを指定します。	必須
db.major_version	データベースのメジャーバージョンを設定します。	Optional
db.minor_version	データベースのマイナーバージョンを設定します。	Optional
db.disable_upsert	データベース upsert を無効にします。	Optional
db.disable_indexing	テーブルインデックスが作成されるかどうかを指定します。	Optional
table.string.table_name_prefix	テーブル名の追加接頭辞を指定します。	Optional
table.string.<id|data|timestamp>.name	列名を指定します。	必須
table.string.<id|data|timestamp>.type	列タイプを指定します。	必須
key_to_string_mapper	**TwoWayKey2StringMapper** クラスを指定します。	Optional



注記

Binary キャッシュストアから古い Data Grid バージョンの移行には、以下のプロパティーで **table.string.*** を **table.binary.*** に変更します。

- **source.table.binary.table_name_prefix**
- **source.table.binary.<id\|data\|timestamp>.name**
- **source.table.binary.<id\|data\|timestamp>.type**

```
# Example configuration for migrating to a JDBC String-Based cache store
target.type=STRING
target.cache_name=myCache
target.dialect=POSTGRES
target.marshaller.class=org.example.CustomMarshaller
target.marshaller.externalizers=25:Externalizer1,org.example.Externalizer2
target.connection_pool.connection_url=jdbc:postgresql:postgres
target.connection_pool.driver_class=org.postrgesql.Driver
target.connection_pool.username=postgres
target.connection_pool.password=redhat
target.db.major_version=9
target.db.minor_version=5
target.db.disable_upsert=false
target.db.disable_indexing=false
target.table.string.table_name_prefix=tablePrefix
target.table.string.id.name=id_column
target.table.string.data.name=datum_column
target.table.string.timestamp.name=timestamp_column
target.table.string.id.type=VARCHAR
target.table.string.data.type=bytea
target.table.string.timestamp.type=BIGINT
target.key_to_string_mapper=org.infinispan.persistence.keymappers.
DefaultTwoWayKey2StringMapper
```

表3.4 RocksDB プロパティー

プロパティー	説明	必須/オプション
location	データベースディレクトリーを設定します。	必須
圧縮	使用する圧縮タイプを指定します。	Optional

```
# Example configuration for migrating from a RocksDB cache store.
source.type=ROCKSDB
source.cache_name=myCache
source.location=/path/to/rocksdb/database
source.compression=SNAPPY
```

表3.5 SingleFileStore プロパティー

プロパティー	説明	必須/オプション
location	キャッシュストア **.dat** ファイルが含まれるディレクトリーを設定します。	必須

```
# Example configuration for migrating to a Single File cache store.
target.type=SINGLE_FILE_STORE
target.cache_name=myCache
target.location=/path/to/sfs.dat
```

表3.6 SoftIndexFileStore プロパティー

プロパティー	説明	値
必須/オプション	**location**	データベースディレクトリーを設定します。
必須	**index_location**	データベースインデックスディレクトリーを設定します。

```
# Example configuration for migrating to a Soft-Index File cache store.
target.type=SOFT_INDEX_FILE_STORE
target.cache_name=myCache
target.location=path/to/sifs/database
target.location=path/to/sifs/index
```

3.4. DATA GRID キャッシュストアの移行

StoreMigrator を実行して、あるキャッシュストアから別のキャッシュストアにデータを移行します。

前提条件

- **infinispan-tools.jar** を取得します。
- ソースおよびターゲットのキャッシュストアを設定する **migrator.properties** ファイルを作成します。

手順

- ソースから **infinispan-tools.jar** をビルドする場合は、以下を実行します。
  1. JDBC ドライバーなどのソースおよびターゲットのデータベースの **infinispan-tools.jar** および依存関係をクラスパスに追加します。
  2. **migrator.properties** ファイルを **StoreMigrator** の引数として指定します。
- Maven リポジトリーから **infinispan-tools.jar** をプルする場合は、以下のコマンドを実行します。
  mvn exec:java