



Red Hat Integration 2022.Q2

Debezium の RHEL へのインストール

Red Hat Enterprise Linux (RHEL) での Debezium 1.7 の使用

Red Hat Integration 2022.Q2 Debezium の RHEL へのインストール

Red Hat Enterprise Linux (RHEL) での Debezium 1.7 の使用

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本ガイドでは、AMQ Streams を使用して RHEL に Red Hat Debezium をインストールする方法を説明します。

目次

| | |
|--|---|
| はじめに | 3 |
| 多様性を受け入れるオープンソースの強化 | 3 |
| 第1章 DEBEZIUM の概要 | 4 |
| 第2章 DEBEZIUM コネクターの RHEL へのインストール | 5 |
| 2.1. 前提条件 | 5 |
| 2.2. KAFKA トピック作成に関する推奨事項 | 5 |
| 2.3. AMQ STREAMS ON RHEL での DEBEZIUM のデプロイ | 6 |
| 付録A サブスクリプションの使用 | 8 |
| アカウントへのアクセス | 8 |
| サブスクリプションのアクティベート | 8 |
| zip および tar ファイルのダウンロード | 8 |

はじめに

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](#) をご覧ください。

第1章 DEBEZIUM の概要

Debezium for Red Hat Integration は、データベース操作をキャプチャーし、行レベル操作のデータ変更イベントレコードを作成して、Apache Kafka トピックに変更イベントレコードをストリーミングする分散型プラットフォームです。Debezium は Apache Karaf に構築され、AMQ Streams とデプロイおよび統合されます。

Debezium によって、データベーステーブルの行レベルの変更がキャプチャーされ、対応する変更イベントが AMQ Streams に渡されます。アプリケーションはこれらの **変更イベントストリーム** を読み取りでき、変更イベントが発生した順にアクセスできます。

[Debezium](#) は、Debezium for Red Hat Integration のアップストリームコミュニティプロジェクトです。

Debezium には、以下を含む複数の用途があります。

- データレプリケーション
- キャッシュの更新およびインデックスの検索
- モノリシックアプリケーションの簡素化
- データ統合
- ストリーミングクエリーの有効化

Debezium は、以下の共通データベースのコネクタ (Kafka Connect をベースとする) を提供します。

- [Db2](#)
- [MySQL](#)
- [MongoDB](#)
- [Oracle](#) (テクノロジープレビュー)
- [PostgreSQL](#)
- [SQL Server](#)

第2章 DEBEZIUM コネクターの RHEL へのインストール

コネクタプラグインで Kafka Connect を拡張して、AMQ Streams 経由で Debezium コネクターをインストールします。AMQ Streams のデプロイ後に、Kafka Connect で Debezium をコネクタ設定としてデプロイできます。

2.1. 前提条件

Debezium のインストールには、以下が必要です。

- Red Hat Enterprise Linux が実行しています。
- 管理者権限 (**sudo** アクセス)。
- AMQ Streams 2.0 on Red Hat Enterprise Linux がホストコンピューターにインストールされている。
- AMQ Streams のインストール時に作成された **kafka** ユーザーのクレデンシャル。
- AMQ Streams クラスターが稼働している。
 - 単一の ZooKeeper ノードおよび単一の Kafka ノードが含まれる実稼働ではない、基本的な AMQ Streams クラスターを実行するための手順は、[Running a single node AMQ Streams cluster](#) を参照してください。



注記

以前のバージョンの AMQ Streams を使用している場合は、最初に AMQ Streams 2.0 にアップグレードする必要があります。アップグレードの手順は、[AMQ Streams and Kafka upgrades](#) を参照してください。

関連情報

- Red Hat Enterprise Linux で Debezium を実行するためにサポートされている設定については、[Debezium Supported Configurations page](#) を参照してください。
- AMQ Streams のインストール方法については、[Installing AMQ Streams](#) を参照してください。

2.2. KAFKA トピック作成に関する推奨事項

Debezium は、データを複数の Apache Kafka トピックに保存します。トピックは、管理者が事前に作成する必要があります。または、Kafka Connect を設定して [トピックを自動的に設定します](#)。

以下のリストで、トピックの作成時に考慮すべき制限および推奨事項を説明します。

MySQL、SQL Server、Db2、および Oracle コネクターのデータベース履歴トピック

- 無限または非常に長い保持期間。
- 3 以上の実稼働環境でのレプリケーション係数
- 単一パーティション

その他のトピック

- 指定のレコードの **最後**の変更イベントのみが保存されるように [Kafka ログコンパクション](#) を有効にする場合は、Apache Kafka で以下のトピックプロパティを設定します。
 - **min.compaction.lag.ms**
 - **delete.retention.ms**
トピックコンシューマーがすべてのイベントを受信してマーカを削除するのに十分な時間を確保するには、シンクコネクタに予想される最大ダウンタイムよりも大きい値を前述のプロパティの値に指定します。たとえば、シンクコネクタに更新を適用する際に発生する可能性のあるダウンタイムを考慮してください。
- 実稼働でレプリケート
- 単一パーティション
単一パーティションルールを緩和できますが、アプリケーションはデータベースの異なる行に対して順不同のイベントを処理する必要があります。単一行のイベントは、引き続き完全に順序付けされます。複数のパーティションを使用する場合は、Kafka がキーをハッシュ化してパーティションを決定するのがデフォルトの動作になります。その他のパーティションストラテジーでは、単一メッセージ変換 (SMT: Single Message Transformations) を使用して、各レコードにパーティション番号を設定する必要があります。

2.3. AMQ STREAMS ON RHEL での DEBEZIUM のデプロイ

この手順では、Red Hat Enterprise Linux で Debezium のコネクタを設定する方法を説明します。コネクタは、Apache Kafka Connect を使用して AMQ Streams クラスタにデプロイされます。Kafka Connect は Apache Kafka と外部システムとの間でデータをストリーミングするためのフレームワークです。Kafka Connect は、スタンドアロンモードではなく分散モードで実行する必要があります。

この手順では、AMQ Streams がインストールされ、ZooKeeper および Apache Kafka が稼働していることを仮定します。

手順

1. Red Hat カスタマーポータル [の Red Hat Integration のダウンロードサイト](#) にアクセスし、Debezium コネクタまたは使用するコネクタをダウンロードします。たとえば、**Debezium 1.7 MySQL Connector** をダウンロードして、Debezium を MySQL データベースと使用します。
2. `/opt/kafka` で、他の Kafka Connect プラグインに対してディレクトリが作成されていない場合は、**connector-plugins** ディレクトリを作成します。

```
$ sudo mkdir /opt/kafka/connector-plugins
```

3. Debezium コネクタアーカイブの内容を `/opt/kafka/connector-plugins` ディレクトリにデプロイメントします。
この例では、MySQL コネクタの内容がデプロイメントされます。

```
$ sudo unzip debezium-connector-mysql-1.7.2.Final.zip -d /opt/kafka/connector-plugins
```

4. インストールするコネクタごとに、前述の手順を繰り返します。
5. **kafka** ユーザーに切り替えます。

```
$ su - kafka
$ Password:
```

6. Kafka Connect プロセスが実行中の場合は、停止します。
 - a. 次のコマンドを入力して、Kafka Connect が分散モードで実行されているかどうかを確認します。

```
$ jcmd | grep ConnectDistributed
```

プロセスが実行中の場合、コマンドはプロセス ID を返します。次に例を示します。

```
18514 org.apache.kafka.connect.cli.ConnectDistributed /opt/kafka/config/connect-distributed.properties
```

- b. プロセス ID を指定して **kill** コマンドを入力し、プロセスを停止します。次に例を示します。

```
$ kill 18514
```

7. **/opt/kafka/config/** の **connect-distributed.properties** ファイルを編集し、Debezium コネクターの場所を指定します。

```
plugin.path=/opt/kafka/connector-plugins
```

8. 分散モードで Kafka Connect を起動します。

```
$ /opt/kafka/bin/connect-distributed.sh /opt/kafka/config/connect-distributed.properties
```

Kafka Connect が実行されます。起動中に、Debezium コネクターは **connector-plugins** ディレクトリーからロードされます。

9. 各 Kafka Connect ワーカーノードに対して 6 から 8 を繰り返し行います。

関連情報

- [Kafka Connect in distributed mode](#)
- [Adding connector plugins](#)

Kafka Connect の更新

デプロイメントを更新する必要がある場合は、**/opt/kafka/connector-plugins** ディレクトリーの Debezium コネクター JAR ファイルを修正してから、Kafka Connect を再起動します。

次のステップ

Debezium User Guideには、変更データキャプチャーに各コネクターとそのソースデータベースを設定する方法が記載されています。設定が終了したら、コネクターはソースデータベースに接続し、挿入、更新、削除された各行または各ドキュメントのイベントを生成します。

付録A サブスクリプションの使用

Integration は、ソフトウェアサブスクリプションを通じて提供されます。サブスクリプションを管理するには、Red Hat カスタマーポータルでアカウントにアクセスします。

アカウントへのアクセス

1. access.redhat.com に移動します。
2. アカウントがない場合は作成します。
3. アカウントにログインします。

サブスクリプションのアクティベート

1. access.redhat.com に移動します。
2. **My Subscriptions** に移動します。
3. **Activate a subscription** に移動し、16 桁のアクティベーション番号を入力します。

zip および tar ファイルのダウンロード

zip または tar ファイルにアクセスするには、カスタマーポータルを使用して、ダウンロードする関連ファイルを検索します。RPM パッケージを使用している場合、この手順は必要ありません。

1. ブラウザーを開き、access.redhat.com/downloads で Red Hat カスタマーポータルの **Product Downloads** ページにログインします。
2. **INTEGRATION AND AUTOMATION** まで下方方向にスクロールします。
3. **Red Hat Integration** をクリックして、Red Hat Integration ダウンロードページを表示します。
4. コンポーネントの **Download** リンクをクリックします。

2023-02-03 17:41:30 UTC に改訂