



Red Hat Integration 2020.q1

Debezium の RHEL へのインストール

テクノロジープレビュー - Red Hat Enterprise Linux(RHEL)上の Debezium 1.0 向け

Red Hat Integration 2020.q1 Debezium の RHEL へのインストール

テクノロジープレビュー - Red Hat Enterprise Linux(RHEL)上の Debezium 1.0 向け

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Installing_Debezium_on_RHEL.ent file |.

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

前書き	3
第1章 DEBEZIUM の概要	4
1.1. 本書の表記慣例	4
第2章 DEBEZIUM コネクタのインストール	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

前書き



重要

テクノロジープレビューの機能は、Red Hat の実稼働環境のサービスレベルアグリーメント (SLA) ではサポートされず、機能的に完全ではないことがあるため、Red Hat はテクノロジープレビュー機能を実稼働環境に実装することは推奨しません。テクノロジープレビュー機能は、最新の技術をいち早く提供し、開発段階で機能のテストやフィードバックの収集を可能にするために提供されます。サポート範囲の詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

第1章 DEBEZIUM の概要

Red Hat Debezium は、データベースを監視し、変更イベントストリームを作成する分散プラットフォームです。Red Hat 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、発生順に変更イベントにアクセスできます。

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ー (Kafka Connect をベースとする) を提供します。

- MySQL
- PostgreSQL
- SQL Server
- MongoDB



注記

本ガイドでは、Debezium のドキュメントを参照します。Debezium は Debezium のアップストリームコミュニティプロジェクトです。

1.1. 本書の表記慣例

置き換え可能なテキスト

本書では、置き換え可能なテキストは等幅フォントおよびイタリック体で記載されています。

たとえば、以下のコードでは **my-namespace** を namespace の名前に置き換えます。

```
sed -i 's/namespace: ./namespace: my-namespace/' install/cluster-operator/*RoleBinding*.yaml
```

第2章 DEBEZIUM コネクタのインストール

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。

- x86_64 アーキテクチャの Red Hat Enterprise Linux バージョン 7.x または 8.x。
- 管理者権限 (**sudo** アクセス)。
- AMQ Streams 1.4 on Red Hat Enterprise Linux がホストマシンにインストールされている。
 - AMQ Streams は、[サポートされる JVM バージョン](#) の1つで実行する必要があります。
- 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 クラスタが稼働している。
 - 単一の ZooKeeper 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、AMQ Streams 1.4 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 および Kafka のアップグレード](#)」を参照してください。

その他のリソース

- 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、データの保存に複数の Kafka トピックを使用します。トピックは管理者が作成するか、**auto.create.topics.enable** ブローカー設定を使用してトピックの自動作成を有効にして Kafka 自体によって作成される必要があります。

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。

データベース履歴トピック (MySQL および SQL Server コネクタ用)

- 無限 (または非常に長期の保持)
- 3 以上の実稼働環境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

- 必要に応じて、[ログコンパクション](#) を有効にする（指定のレコードの **最後** の変更イベントのみを維持する必要がある場合）。
この場合、Apache Kafka の **min.compaction.lag.ms** および **delete.retention.ms** トピックレベルの設定を設定し、コンシューマーがすべてのイベントを受信し、マーカを削除するのに十分な時間を確保する必要があります。具体的には、これらの値は、シンクコネクタが予想される最大ダウンタイムよりも大きくする必要があります（例：更新する場合など）。
- 実稼働でレプリケートされます。
- 単一パーティション。
単一パーティションルールを緩和することができますが、アプリケーションはデータベースの異なる行の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が使用される場合、Kafka はデフォルトでキーをハッシュ化してパーティションを決定します。他のパーティションストラテジーでは、SMT(Simple Message Transforms)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、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 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 1.0.0 MySQL Connector** をダウンロードして、MySQL データベースで Debezium を使用します。
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` ディレクトリに展開します。
この例では、MySQL コネクタの内容が展開されます。

```
$ sudo unzip debezium-connector-mysql-1.0.0-plugin.zip -d /opt/kafka/connector-plugins
```

4. インストールするコネクタごとに、上記のステップを繰り返し行います。
5. **kafka** ユーザーに切り替えます。

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

6. Kafka Connect がすでに分散モードで稼働しているかどうかを確認します。稼働していれば、すべての Kafka Connect ワーカーノード上で関連プロセスを停止します。以下に例を示します。

```
$ jcmd | grep ConnectDistributed
18514 org.apache.kafka.connect.cli.ConnectDistributed /opt/kafka/config/connect-
distributed.properties
$ 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 を実行します。

```
$ /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。設定の終了後、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

Debezium は、ソフトウェアサブスクリプションから提供されます。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. **サブスクリプション** 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は、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 **ダウンロードリンク** をクリックします。

改訂日時： 2022-07-30 10:01:26 +1000