



Red Hat Integration 2020.q1

Debezium の OpenShift へのインストール

OpenShift Container Platform 上で Debezium 1.0 を使用

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

OpenShift Container Platform 上で Debezium 1.0 を使用

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_OpenShift.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 OpenShift Container Platform に Red Hat Debezium をインストールする方法を説明します。

目次

第1章 DEBEZIUM の概要	3
1.1. 本書の表記慣例	3
第2章 DEBEZIUM コネクターのインストール	4
2.1. 前提条件	4
2.2. KAFKA トピック作成に関する推奨事項	4
2.3. AMQ STREAMS での DEBEZIUM のデプロイ	5
Kafka Connect の更新	7
デプロイメントの確認	7
第3章 KAFKA CONNECT ベースイメージからのコンテナイメージの作成	9
付録A サブスクリプションの使用	11
アカウントへのアクセス	11
サブスクリプションのアクティベート	11
Zip および Tar ファイルのダウンロード	11

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 コネクターのインストール

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。

- OpenShift クラスター
- Kafka Connect S2I での AMQ Streams のデプロイメント
- 必要なクラスターロールおよび API サービスを設定するための **cluster-admin** 権限を持つ OpenShift クラスターのユーザー。



注記

Debezium コネクターを実行するには、Java 8 以降が必要です。

Debezium をインストールするには、OpenShift Container Platform コマンドラインインターフェース (CLI) が必要です。

- OpenShift 3.11 の CLI のインストール方法の詳細は、[OpenShift Container Platform 3.11 のドキュメント](#) を参照してください。
- OpenShift 4.2 の CLI のインストール方法の詳細は、[OpenShift Container Platform 4.2 ドキュメント](#) を参照してください。

関連情報

- AMQ Streams のインストール方法については、「[Using AMQ Streams on OpenShift](#)」を参照してください。
- AMQ Streams には、Kafka コンポーネントをデプロイおよび管理する **Cluster Operator** が含まれています。AMQ Streams Cluster Operator を使用して Kafka コンポーネントをインストールする方法の詳細は、「[Deploying Kafka Connect to your cluster](#)」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、シンクコネクター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 での DEBEZIUM のデプロイ

この手順では、Red Hat OpenShift Container Platform で Debezium のコネクターを設定する方法を説明します。

作業を開始する前に

OpenShift で Apache Kafka および Kafka Connect を設定するには、[Red Hat AMQ Streams](#) を使用します。AMQ Streams は、Kafka を OpenShift に取り入れる operator およびイメージを提供します。

ここでは、Kafka Connect S2I(Source to Image)をデプロイおよび使用します。S2I は、アプリケーションのソースコードを入力として受け入れ、アセンブルされたアプリケーションをアウトプットとして実行する新規イメージを生成するイメージをビルドするフレームワークです。

S2I がサポートされる Kafka Connect ビルダーイメージは、[registry.redhat.io/amq7/amq-streams-kafka-24:1.4.0](#) イメージの一部として、[Red Hat Container Catalog](#) で提供されます。S2I プロセスは（プラグインおよびコネクターとともに）バイナリを取得して、`/tmp/kafka-plugins/s2i` ディレクトリーに保存します。このディレクトリーから、Kafka Connect デプロイメントとともに使用できる新しい Kafka Connect イメージを作成します。改良されたイメージの使用を開始すると、Kafka Connect は `/tmp/kafka-plugins/s2i` ディレクトリーからサードパーティープラグインをロードします。



注記

Kafka Connect S2I をデプロイおよび使用するのではなく、AMQ Streams Kafka イメージに基づいて新しい **Dockerfile** を作成し、コネクターを含めることができます。

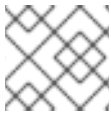
[3章 Kafka Connect ベースイメージからのコンテナイメージの作成](#)を参照してください。

この手順では、以下を行います。

- Kafka クラスターを OpenShift にデプロイします。
- Debezium コネクターのダウンロードおよび設定

- コネクターでの Kafka Connect のデプロイ

Kafka クラスターがすでにデプロイされている場合は、最初の手順を省略できます。



注記

Pod 名は AMQ Streams デプロイメントに対応する必要があります。

手順

1. Kafka クラスターをデプロイします。
 - a. 『[Installing AMQ Streams and deploying components](#)』の手順に従って、AMQ Streams operator をインストールします。
 - b. 希望の設定を選択し、[Kafka Cluster をデプロイ](#) します。
 - c. [Kafka Connect s2i](#) をデプロイします。

これで、Kafka Connect S2I を使用して OpenShift で Kafka クラスターが稼働しました。

2. Pod が実行されていることを確認します。

```
$ oc get pods
```

```
NAME                                READY STATUS
<cluster-name>-entity-operator-7b6b9d4c5f-k7b92  3/3 Running
<cluster-name>-kafka-0                    2/2 Running
<cluster-name>-zookeeper-0                2/2 Running
<cluster-name>-operator-97cd5cf7b-l58bq      1/1 Running
```

Pod の実行の他に、**DeploymentConfig** が Connect **S2I** に関連付けられている必要があります。

3. リリース 1.0 を選択し、[Red Hat Integration のダウンロードサイト](#) からデータベースの [Debezium コネクターアーカイブ](#) をダウンロード します。
4. アーカイブを展開して、コネクタープラグインのディレクトリー構造を作成します。

```
$ tree ./my-plugin/
./my-plugin/
├── debezium-connector-mongodb
│   ├── ...
│   ├── debezium-connector-mysql
│   │   ├── ...
│   │   ├── debezium-connector-postgres
│   │   │   ├── ...
│   │   │   └── debezium-connector-sqlserver
│   │       └── ...
└── ...
```

これで Kafka Connect S2I ビルドがトリガーされます。

5. ビルド設定の名前を確認します。

```
$ oc get buildconfigs
```

NAME	TYPE	FROM	LATEST
<cluster-name>-cluster-connect	Source	Binary	2

6. **oc start-build** コマンドを使用して、Debezium ディレクトリーを使用して Kafka Connect イメージの新しいビルドを開始します。

```
oc start-build <cluster-name>-cluster-connect --from-dir ./my-plugin/
```



注記

ビルドの名前は、デプロイされた Kafka Connect クラスターと同じになります。

7. 更新されたデプロイメントが実行していることを確認します。

```
oc get pods
```

NAME	READY	STATUS
<cluster-name>-entity-operator-7b6b9d4c5f-k7b92	3/3	Running
<cluster-name>-kafka-0	2/2	Running
<cluster-name>-zookeeper-0	2/2	Running
<cluster-name>-cluster-connect-2-jw695	1/1	Running
<cluster-name>-cluster-connect-2-deploy	0/1	Completed
strimzi-cluster-operator-97cd5cf7b-l58bq	1/1	Running

または、OpenShift Web コンソールの **Pods** ビューに移動して、Pod が実行中であることを確認できます。

NAME	NAMESPACE	POD LABELS	NODE	STATUS	READINESS
debezium-kafka-cluster-entity-operator-5dbdcd6fdb-z6qcv	debezium	pod-template-hash=5dbdcd6fdb strimzi.io/c...=debezium-kafka-... strimzi.io/kind=Kafka strimzi...=debezium-kafka-clus...	ip-10-0-151-231.eu-west-1.compute.internal	Running	Ready
debezium-kafka-cluster-kafka-0	debezium	controlle...=debezium-kafka-c... statefulset.kub...=debezium-ka... strimzi.io/c...=debezium-kafka-... strimzi.io/kind=Kafka strimzi...=debezium-kafka-cl...	ip-10-0-135-247.eu-west-1.compute.internal	Running	Ready
debezium-kafka-cluster-zookeeper-0	debezium	controlle...=debezium-kafka-cl... statefulset.ku...=debezium-ka... strimzi.io/c...=debezium-kafka-... strimzi.io/kind=Kafka strimzi...=debezium-kafka-clu...	ip-10-0-167-205.eu-west-1.compute.internal	Running	Ready
debezium-kafka-connect-cluster-connect-2-jw695	debezium	depl...=debezium-kafka-conne... deploy...=debezium-kafka-co... strimzi...=debezium-kafka-c... strimzi.io/kind=KafkaConnectS2I strimzi...=debezium-kafka-con...	ip-10-0-167-205.eu-west-1.compute.internal	Running	Ready
strimzi-cluster-operator-97cd5cf7b-l58bq	debezium	name=strimzi-cluster-operator pod-template-hash=97cd5cf7b strimzi.io/kind=cluster-operator	ip-10-0-151-231.eu-west-1.compute.internal	Running	Ready

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、Debezium ディレクトリーの JAR ファイルを修正し、Kafka Connect を再ビルドします。

デプロイメントの確認

ビルドが完了したら、Kafka Connect のデプロイメントによって新しいイメージが自動的に使用されます。

コネクタが起動すると、ソースに接続し、挿入、更新、削除された各行またはドキュメントのイベントを生成します。

『[Debezium スタートガイド](#)』の手順に従ってデプロイメントが正しいことを確認します。[サービスの起動](#)には、Kafka クラスターの設定および Kafka Connect の設定手順が記載されています。

特定のコネクタを設定するには、以下を参照してください。

- [Deploying the MySQL connector](#)
- [「Deploying the MongoDB connector」](#)
- [「Deploying the PostgreSQL connector」](#)
- [Deploying the SQL Server connector](#)

第3章 KAFKA CONNECT ベースイメージからのコンテナイメージの作成

Kafka Connect S2I を使用する代わりに、Docker を使用して独自の CDC イメージを構築する方法があります。[Red Hat Container Catalog](#) の Kafka コンテナイメージを、追加のコネクタプラグインで独自のカスタムイメージを作成するためのベースイメージとして使用できます。

以下の手順では、カスタムイメージを作成し、`/opt/kafka/plugins` ディレクトリーに追加する方法を説明します。起動時に、Debezium バージョンの Kafka Connect は `/opt/kafka/plugins` ディレクトリーに含まれるサードパーティーのコネクタプラグインをロードします。

前提条件

- AMQ Streams Cluster Operator がデプロイされている必要があります。

手順

1. `registry.redhat.io/amq7/amq-streams-kafka-24:1.4.0` をベースイメージとして使用して、新規の `Dockerfile` を作成します。

```
FROM registry.redhat.io/amq7/amq-streams-kafka-24:1.4.0
USER root:root
COPY ./my-plugins/ /opt/kafka/plugins/
USER 1001
```

2. コンテナイメージをビルドします。

```
docker build -t my-new-container-image:latest
```

3. カスタムイメージをコンテナレジストリーにプッシュします。

```
docker push my-new-container-image:latest
```

4. 新しいコンテナイメージを示します。
以下のいずれかを行います。

- **KafkaConnect** カスタムリソースの `KafkaConnect.spec.image` プロパティを編集します。
このプロパティが設定されていると、クラスターオペレータの `STRIMZI_DEFAULT_KAFKA_CONNECT_IMAGE` 変数がオーバーライドされます。

```
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaConnect
metadata:
  name: my-connect-cluster
spec:
  #...
  image: my-new-container-image
```

または

- `install/cluster-operator/050-Deployment-strimzi-cluster-operator.yaml` ファイルの `STRIMZI_DEFAULT_KAFKA_CONNECT_IMAGE` 変数を編集し、新しいコンテナイ

メッセージを示すようにした後、Cluster Operator を再インストールします。このファイルを編集する場合は、これを OpenShift クラスターに適用する必要があります。

関連情報

- **KafkaConnect.spec.image property** および **STRIMZI_DEFAULT_KAFKA_CONNECT_IMAGE** 変数の詳細は、[Using AMQ Streams on OpenShift](#) を参照してください。

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 10:01:05 +1000