Red Hat Streams for Apache Kafka 2.7

OpenShift 上の Streams for Apache Kafka スタートガイド

OpenShift Container Platform で Streams for Apache Kafka 2.7 を使い始める

Last Updated: 2024-09-16

Red Hat Streams for Apache Kafka 2.7 OpenShift 上の Streams for Apache Kafka スタートガイド

OpenShift Container Platform で Streams for Apache Kafka 2.7 を使い始める

法律上の通知

概要

OpenShift 上に Kafka クラスターを作成して、Streams for Apache Kafka を試用します。Kafka クラスターに接続し、Kafka トピックからメッセージを送受信します。

目次

はじめに

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。

改善を提案するには、Jira 課題を作成し、変更案についてご説明ください。ご要望に迅速に対応できるよう、できるだけ詳細にご記入ください。

前提条件

- Red Hat カスタマーポータルのアカウントがある。このアカウントを使用すると、Red Hat Jira Software インスタンスにログインできます。
  アカウントをお持ちでない場合は、アカウントを作成するように求められます。

手順

1. 以下の Create issue をクリックします。
2. **Summary** テキストボックスに、問題の簡単な説明を入力します。
3. **Description** テキストボックスに、次の情報を入力します。
   - 問題が見つかったページの URL
   - 問題の詳細情報
     他のフィールドの情報はデフォルト値のままにすることができます。
4. レポーター名を追加します。
5. **Create** をクリックして、Jira 課題をドキュメントチームに送信します。

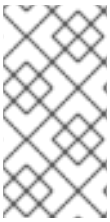
フィードバックをご提供いただきありがとうございました。

第1章 概要

Red Hat Streams for Apache Kafka を使用して Kafka クラスターを作成およびセットアップし、アプリケーションとサービスをそのクラスターに接続します。

このガイドでは、OpenShift Container Platform に Streams for Apache Kafka をインストールして使用を開始する方法を説明します。OpenShift Web コンソールの OperatorHub から Streams for Apache Kafka Operator を直接インストールできます。Streams for Apache Kafka Operator は、Kafka コンポーネントのインストールおよび管理方法を把握しています。OperatorHub からインストールすると、自動更新を利用できる Streams for Apache Kafka の標準設定が提供されます。

Streams for Apache Kafka Operator をインストールすると、Kafka コンポーネントのインスタンスをインストールするためのリソースが提供されます。Kafka クラスターをインストールした後、メッセージの生成と消費を開始できます。



注記

さらに柔軟なデプロイが必要な場合は、Streams for Apache Kafka で提供されるインストールアーティファクトを使用できます。インストールアーティファクトの使用に関する詳細は、**OpenShift での Streams for Apache Kafka のデプロイと管理**を参照してください。

1.1. 前提条件

Streams for Apache Kafka を使い始めるには、次の前提条件を満たす必要があります。

- Red Hat アカウントを持っている。
- JDK 11 以降がインストールされている。
- OpenShift 4.12 ～ 4.15 クラスターを使用できる。
- OpenShift **oc** コマンドラインツールがインストールされ、稼働中のクラスターに接続するように設定されている。

開始する手順は、OpenShift Web コンソールの OperatorHub の使用に基づいていますが、**OpenShift** oc CLI ツールを使用して特定の操作を実行することもできます。**oc** ツールを使用して OpenShift クラスターに接続する必要があります。

- **'?'** ヘルプメニュー、**Command Line Tools** の順にクリックすると、Web コンソールから **oc** CLI ツールをインストールできます。
- プロファイル名をクリックしてから **Copy login command** をクリックすると、Web コンソールから必要な **oc login** の詳細をコピーできます。

1.2. 関連情報

- Strimzi の概要
- OpenShift 上の Streams for Apache Kafka のデプロイとアップグレード

第2章 OPERATORHUB からの STREAMS FOR APACHE KAFKA OPERATOR のインストール

OpenShift Container Platform Web コンソールの OperatorHub を使用して、Streams for Apache Kafka Operator をインストールし、サブスクライブすることができます。

この手順では、プロジェクトを作成し、そのプロジェクトに Streams for Apache Kafka Operator をインストールする方法を説明します。プロジェクトは namespace を表します。namespace を使用して機能を分離することで管理性を確保することを推奨します。

警告

必ず適切な更新チャネルを使用してください。サポート対象の OpenShift のバージョンを使用している場合は、デフォルトの stable チャネルから安全に Streams for Apache Kafka をインストールできます。ただし、stable チャネルで自動更新を有効にすることは推奨されません。自動アップグレードでは、アップグレード前の必要手順がスキップされます。バージョン固有のチャネルでのみ自動アップグレードを使用します。

前提条件

- **cluster-admin** または **strimzi-admin** パーミッションを持つアカウントを使用して OpenShift Container Platform Web コンソールにアクセスできる。

手順

1. OpenShift Web コンソールで **Home > Projects** ページに移動し、インストール用のプロジェクト (namespace) を作成します。
   この例では、**amq-streams-kafka** という名前のプロジェクトを使用します。
2. **Operators > OperatorHub** ページに移動します。
3. **Filter by keyword** ボックスにスクロールするかキーワードを入力して、**Streams for Apache Kafka** Operator を見つけます。
   Operator は、**Streaming & Messaging** カテゴリーにあります。
4. **Streams for Apache Kafka** をクリックして、Operator の情報を表示します。
5. operator に関する情報を確認し、**Install** をクリックします。
6. **Install Operator** ページで、次のインストールおよび更新オプションから選択します。
   - **Update Channel**: Operator の更新チャネルを選択します。
     - **stable** チャネル (デフォルト) には最新の更新とリリースがすべて含まれます。これには、十分なテストを行った上、安定していることが想定される、メジャー、マイナー、およびマイクロリリースが含まれます。
     - **amq-streams-X.x** チャネルには、メジャーリリースのマイナーリリースの更新およびマイクロリリースの更新が含まれます。**X** は、メジャーリリースのバージョン番号に置き換えてください。

    - **amq-streams-X.Y.x** チャネルには、マイナーリリースのマイクロリリースの更新が含まれます。**X** はメジャーリリースのバージョン番号、**Y** はマイナーリリースのバージョン番号に置き換えてください。

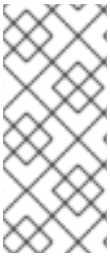
- **Installation Mode**: 作成したプロジェクトを選択して、特定の namespace に Operator をインストールします。
  Streams for Apache Kafka Operator は、クラスター内のすべての namespace (デフォルトのオプション) か、特定の namespace にインストールできます。Kafka クラスターとその他の Streams for Apache Kafka コンポーネントに特定の namespace を割り当てることを推奨します。

- **Update approval**: デフォルトでは、Streams for Apache Kafka Operator は、Operator Lifecycle Manager (OLM) によって最新の Streams for Apache Kafka バージョンに自動的にアップグレードされます。今後のアップグレードを手動で承認する場合は、**Manual** を選択します。Operator の詳細は、OpenShift ドキュメント を参照してください。

7. **Install** をクリックして、選択した namespace に Operator をインストールします。
   Streams for Apache Kafka Operator は、選択した namespace に Cluster Operator、CRD、およびロールベースアクセス制御 (RBAC) リソースをデプロイします。

8. Operator を使用する準備ができたら、**Operators > Installed Operators** に移動して、Operator が選択した namespace にインストールされていることを確認します。
   ステータスは **Succeeded** と表示されます。

   これで、Streams for Apache Kafka Operator を使用して、Kafka クラスターから順に Kafka コンポーネントをデプロイできるようになりました



注記

Workloads > Deployments に移動すると、Cluster Operator および Entity Operator のデプロイメントの詳細を確認できます。Cluster Operator の名前には、バージョン番号 **amq-streams-cluster-operator-<version>** が含まれています。Streams for Apache Kafka インストールアーティファクトを使用して Cluster Operator をデプロイする場合、名前は異なります。この場合、名前は **strimzi-cluster-operator** です。

第3章 STREAMS FOR APACHE KAFKA OPERATOR を使用した KAFKA コンポーネントのデプロイ

Openshift に Streams for Apache Kafka Operator をインストールすると、この Operator によって、ユーザーインターフェイスから Kafka コンポーネントをインストールできるようになります。

次の Kafka コンポーネントをインストールできます。

- Kafka
- Kafka Connect
- Kafka MirrorMaker
- Kafka MirrorMaker 2
- Kafka Topic
- Kafka User
- Kafka Bridge
- Kafka Connector
- Kafka Rebalance

コンポーネントを選択して、インスタンスを作成します。少なくとも、Kafka インスタンスを作成します。この手順では、デフォルト設定を使用して Kafka インスタンスを作成する方法を説明します。インストールを実行する前に、デフォルトのインストール仕様を設定できます。

プロセスは、他の Kafka コンポーネントのインスタンスを作成する場合と同じです。

前提条件

- Streams for Apache Kafka Operator が OpenShift クラスターにインストール されている。

手順

1. Web コンソールで **Operators > Installed Operators** ページに移動し、**Streams for Apache Kafka** をクリックして Operator の詳細を表示します。
   提供されている API から、Kafka コンポーネントのインスタンスを作成できます。

2. **Kafka** の下の **Create instance** をクリックして、Kafka インスタンスを作成します。
   デフォルトでは、3 つの Kafka ブローカーノードと 3 つの ZooKeeper ノードを持つ **my-cluster** という名の Kafka クラスターを作成します。クラスターはエフェメラルストレージを使用します。

3. **Create** をクリックして、Kafka のインストールを開始します。
   ステータスが **Ready** に変わるまで待ちます。

第4章 KAFKA クラスターにアクセスするための OPENSHIFT ルートの作成

OpenShift の外部で Kafka クラスターにアクセスするための OpenShift ルートを作成します。

この手順では、Kafka クラスターを OpenShift 環境外のクライアントに公開する方法を説明します。Kafka クラスターが公開された後、外部クライアントは Kafka クラスターからのメッセージを生成および消費できます。

OpenShift ルートを作成するために、OpenShift にインストールされている Kafka クラスターの設定に **route** リスナーが追加されます。

警告

OpenShift Route アドレスは、Kafka クラスターの名前、リスナーの名前、および作成される namespace の名前で設定されます。たとえば、**my-cluster-kafka-listener1-bootstrap-amq-streams-kafka** (<cluster_name>-kafka-<**listener_name**>-bootstrap-<**namespace**>) です。アドレスの全体の長さが上限の 63 文字を超えないように注意してください。

前提条件

- OpenShift で Kafka クラスターを作成 している。
- 証明書を管理するには、OpenJDK **keytool** が必要である。
- (オプション) OpenShift **oc** CLI ツールを使用していくつかのステップを実行できる。

手順

1. Web コンソールで **Operator** > **Installed Operator** ページに移動し、**Streams for Apache Kafka** を選択して Operator の詳細を表示します。
2. **Kafka** ページを選択して、インストールされている Kafka クラスターを表示します。
3. 設定している Kafka クラスターの名前をクリックして、その詳細を表示します。
この例では、**my-cluster** という名前の Kafka クラスターを使用します。
4. Kafka クラスター **my-cluster** の **YAML** ページを選択します。
5. ルートリスナー設定を追加して、**listener1** という名前の OpenShift ルートを作成します。
リスナー設定は、**route** タイプに設定する必要があります。Kafka 設定の **listeners** の下にリスナー設定を追加します。

   外部ルートリスナーの設定

```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: my-cluster
```

```
  namespace: amq-streams-kafka
spec:
  kafka:
    # ...
    listeners:
      # ...
      - name: listener1
        port: 9094
        type: route
        tls: true
# ...
```

クライアントはデフォルトのルーターポートであるポート 443 に接続しますが、トラフィックは設定するポート (この例では 9094) にルーティングされます。

6. 更新された設定を保存します。

7. Kafka クラスター **my-cluster** の **Resources** ページを選択して、クライアントに必要な接続情報を見つけます。
Resources ページから、Kafka クラスターに接続するために必要なルートリスナーと公開クラスター証明書の詳細を確認できます。

8. Kafka クラスター用に作成された **my-cluster-kafka-listener1-bootstrap** ルートの名前をクリックして、ルートの詳細を表示します。

9. ホスト名をメモします。
ホスト名は、Kafka クラスターに接続するためのブートストラップアドレスとして、Kafka クライアントのポート 443 で指定されます。

   Networking > Routesに移動し、**amq-streams-kafka** プロジェクトを選択して、namespace に作成されたルートを表示することにより、ブートストラップアドレスを見つけることもできます。

   または、**oc** ツールを使用してブートストラップの詳細を抽出できます。

   ブートストラップ情報の抽出

```
oc get routes my-cluster-kafka-listener1-bootstrap -o=jsonpath='{.status.ingress[0].host}{"\n"}'
```

10. **Resources** ページに戻り、**my-cluster-cluster-ca-cert** の名前をクリックして、Kafka クラスターにアクセスするためのシークレットの詳細を表示します。
ca.crt 証明書ファイルには、Kafka クラスターの公開証明書が含まれています。

    Kafka ブローカーにアクセスするには証明書が必要です。

11. **ca.crt** 公開証明書ファイルのローカルコピーを作成します。
証明書の詳細をコピーするか、OpenShift **oc** ツールを使用してそれらを抽出できます。

    公開証明書の抽出

```
oc extract secret/my-cluster-cluster-ca-cert --keys=ca.crt --to=- > ca.crt
```

12. **keytool** を使用して、公開クラスター証明書のローカルトラストストアを作成します。

    ローカルトラストストアの作成

```
keytool -keystore client.truststore.jks -alias CARoot -import -file ca.crt
```

プロンプトが表示されたら、トラストストアにアクセスするためのパスワードを作成します。

トラストストアは、Kafka クラスターへのアクセスを認証するために Kafka クライアントで指定されます。

メッセージの送受信を開始する準備が整いました。

第5章 トピックからのメッセージの送受信

OpenShift にインストールされている Kafka クラスターとの間でメッセージを送受信します。

この手順では、Kafka クライアントを使用してメッセージを生成および消費する方法を説明します。クライアントを OpenShift にデプロイするか、ローカル Kafka クライアントを OpenShift クラスターに接続できます。いずれかまたは両方のオプションを使用して、Kafka クラスターのインストールをテストできます。ローカルクライアントの場合は、OpenShift ルート接続を使用して Kafka クラスターにアクセスします。

oc コマンドラインツールを使用して、Kafka クライアントをデプロイして実行します。

前提条件

- OpenShift で Kafka クラスターを作成 している。

ローカルプロデューサーおよびコンシューマーの場合:

- OpenShift で実行している Kafka クラスターへの外部アクセス用のルートを作成 している。
- Streams for Apache Kafka ソフトウェアダウンロードページ から最新の Kafka クライアントバイナリーにアクセスできる。

OpenShift クラスターにデプロイされた Kafka クライアントからのメッセージの送受信

プロデューサーおよびコンシューマーのクライアントを OpenShift クラスターにデプロイします。その後、クライアントを使用して、同じ namespace 内の Kafka クラスターとの間でメッセージを送受信できます。このデプロイメントでは、Kafka を実行するために Streams for Apache Kafka コンテナーイメージを使用します。

1. **oc** コマンドラインインターフェイスを使用して、Kafka プロデューサーをデプロイします。この例では、Kafka クラスター **my-cluster** に接続する Kafka プロデューサーをデプロイします。

   my-topic という名前のトピックが作成されます。

   Kafka プロデューサーの OpenShift へのデプロイ

   ```
   oc run kafka-producer -ti \
   --image=registry.redhat.io/amq-streams/kafka-37-rhel9:2.7.0 \
   --rm=true \
   --restart=Never \
   -- bin/kafka-console-producer.sh \
   --bootstrap-server my-cluster-kafka-bootstrap:9092 \
   --topic my-topic
   ```

   注記

   接続に失敗した場合は、Kafka クラスターが実行中で、正しいクラスター名が **bootstrap-server** として指定されていることを確認してください。

2. コマンドプロンプトから、いくつかのメッセージを入力します。
3. OpenShift Web コンソールで **Home** > **Projects** ページに移動し、作成した **amq-streams-kafka** プロジェクトを選択します。

4. Pod のリストから、**kafka-producer** をクリックして、プロデューサー Pod の詳細を表示します。

5. **Logs** ページを選択して、入力したメッセージが存在することを確認します。

6. **oc** コマンドラインインターフェイスを使用して、Kafka コンシューマーをデプロイします。

   Kafka コンシューマーの OpenShift へのデプロイ

   ```
   oc run kafka-consumer -ti \
   --image=registry.redhat.io/amq-streams/kafka-37-rhel9:2.7.0 \
   --rm=true \
   --restart=Never \
   -- bin/kafka-console-consumer.sh \
   --bootstrap-server my-cluster-kafka-bootstrap:9092 \
   --topic my-topic \
   --from-beginning
   ```

   コンシューマーは **my-topic** に生成されたメッセージを消費しました。

7. コマンドプロンプトから、コンシューマーコンソールに着信メッセージが表示されていることを確認します。

8. OpenShift Web コンソールで **Home > Projects** ページに移動し、作成した **amq-streams-kafka** プロジェクトを選択します。

9. Pod のリストから、**kafka-consumer** をクリックして、コンシューマー Pod の詳細を表示します。

10. **Logs** ページを選択して、消費したメッセージが存在することを確認します。

ローカルで実行されている Kafka クライアントからのメッセージの送受信

コマンドラインインターフェイスを使用して、ローカルマシンで Kafka プロデューサーとコンシューマーを実行します。

1. Streams for Apache Kafka ソフトウェアダウンロードページ から **Streams for Apache Kafka<バージョン>** バイナリーをダウンロードして展開します。
   amq-streams-<version>-bin.zip ファイルを任意の場所に解凍します。

2. コマンドラインインターフェイスを開き、トピック **my-topic** と TLS の認証プロパティーを使用して Kafka コンソールプロデューサーを起動します。
   OpenShift ルートを使用して Kafka ブローカーにアクセスする のに必要なプロパティーを追加します。

   - 使用している OpenShift ルートのホスト名およびポート 443 を使用します。
   - パスワードと、ブローカー証明書用に作成したトラストストアへの参照を使用します。

     ローカル Kafka プロデューサーの起動

     ```
     kafka-console-producer.sh \
     --bootstrap-server my-cluster-kafka-listener1-bootstrap-amq-streams-kafka.apps.ci-ln-50kcyvt-72292.origin-ci-int-gce.dev.rhcloud.com:443 \
     --producer-property security.protocol=SSL \
     ```

```
--producer-property ssl.truststore.password=password \
--producer-property ssl.truststore.location=client.truststore.jks \
--topic my-topic
```

3. プロデューサーが実行しているコマンドラインインターフェイスにメッセージを入力します。
4. Enter を押してメッセージを送信します。
5. 新しいコマンドラインインターフェイスタブまたはウィンドウを開き、Kafka コンソールコンシューマーを起動してメッセージを受信します。
   プロデューサーと同じ接続の詳細を使用します。

   ローカル Kafka コンシューマーの起動

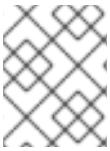
```
kafka-console-consumer.sh \
--bootstrap-server my-cluster-kafka-listener1-bootstrap-amq-streams-kafka.apps.ci-ln-50kcyvt-72292.origin-ci-int-gce.dev.rhcloud.com:443 \
--consumer-property security.protocol=SSL \
--consumer-property ssl.truststore.password=password \
--consumer-property ssl.truststore.location=client.truststore.jks \
--topic my-topic --from-beginning
```

6. コンシューマーコンソールに受信メッセージが表示されることを確認します。
7. Crtl+C を押して、Kafka コンソールプロデューサーとコンシューマーを終了します。

第6章 (プレビュー) STREAMS FOR APACHE KAFKA CONSOLE のデプロイ

Streams for Apache Kafka によって管理される Kafka クラスターをデプロイした後、Streams for Apache Kafka Console をデプロイしてクラスターを接続できます。Streams for Apache Kafka Console は、Kafka クラスターの管理を容易にし、ユーザーインターフェイスから各クラスターを監視、管理、最適化するためのリアルタイムの分析情報を提供します。

Streams for Apache Kafka Console への接続と使用の詳細は、Streams for Apache Kafka ドキュメントのコンソールガイドを参照してください。



注記

Streams for Apache Kafka Console は現在、テクノロジープレビューとして利用できます。

付録A サブスクリプションの使用

Streams for Apache Kafka 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** カテゴリーで、**Streams for Apache Kafka** エントリーを見つけます。
3. 必要な Streams for Apache Kafka 製品を選択します。**Software Downloads** ページが開きます。
4. コンポーネントの **Download** リンクをクリックします。

DNF を使用したパッケージのインストール

パッケージとすべてのパッケージ依存関係をインストールするには、以下を使用します。

```
dnf install <package_name>
```

ローカルディレクトリーからダウンロード済みのパッケージをインストールするには、以下を使用します。

```
dnf install <path_to_download_package>
```

改訂日時: 2024-06-25