Red Hat AMQ 7.7

AMQ Streams on OpenShift の評価

OpenShift Container Platform 上で AMQ Streams 1.5 を使用

Last Updated: 2023-02-03

Red Hat AMQ 7.7 AMQ Streams on OpenShift の評価

OpenShift Container Platform 上で AMQ Streams 1.5 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 © 2023 | You need to change the HOLDER entity in the en-US/Evaluating_AMQ_Streams_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 をインストールおよび管理し、本番環境での潜在的な使用を評価する方法を説明します。

目次

第1章 AMQ STREAMS の概要 3
1.1. KAFKA の機能 3
1.2. KAFKA のユースケース 3
1.3. AMQ STREAMS による KAFKA のサポート 4
1.4. OPERATOR 4
1.5. 本書の表記慣例 5

第2章 AMQ STREAMS の評価 6
2.1. 前提条件 6
2.2. AMQ STREAMS のダウンロード 6
2.3. AMQ STREAMS のインストール 6
2.4. クラスターの作成 8
2.5. クラスターへのアクセス 9
2.6. トピックからのメッセージの送受信 10

付録A サブスクリプションの使用 12
アカウントへのアクセス 12
サブスクリプションのアクティベート 12
Zip および Tar ファイルのダウンロード 12

第1章 AMQ STREAMS の概要

AMQ Streams は、OpenShift クラスターで Apache Kafka を実行するプロセスを簡素化します。

本ガイドでは、AMQ Streams の本番環境を評価する手順を説明します。手順では、AMQ Streams デプロイメントをできるだけ早く稼働する方法を説明します。

AMQ Streams を試す前に、その機能と使用方法を理解しておくと便利です。本章では、Kafka の主な概念を紹介し、AMQ Streams Operator の概要についても取り上げます。

Operator は、OpenShift アプリケーションのパッケージ化、デプロイメント、および管理を行う方法です。AMQ Streams Operator は OpenShift の機能を拡張し、Kafka デプロイメントに関連する共通タスクや複雑なタスクを自動化します。Kafka 操作の情報をコードに実装することで、Kafka の管理タスクは簡素化され、必要な手動の作業が少なくなります。

1.1. KAFKA の機能

Kafka の基盤のデータストリーム処理機能とコンポーネントアーキテクチャーによって以下が提供されます。

- スループットが非常に高く、レイテンシーが低い状態でデータを共有するマイクロサービスおよびその他のアプリケーション
- メッセージの順序の保証
- アプリケーションの状態を再構築するためにデータストレージからメッセージを巻き戻し/再生
- キーバリューログの使用時に古いレコードを削除するメッセージ圧縮
- クラスター設定での水平スケーラビリティー
- 耐障害性を制御するデータのレプリケーション
- 即時アクセス用の大量データの保持

1.2. KAFKA のユースケース

Kafka の機能は、以下に適しています。

- イベント駆動型のアーキテクチャー
- アプリケーションの状態に加えられた変更をイベントのログとしてキャプチャーするイベントソーシング
- メッセージのブローカー
- Web サイトアクティビティーの追跡
- メトリクスによる運用上のモニターリング
- ログの収集および集計
- 分散システムのコミットログ
- アプリケーションがリアルタイムでデータに対応できるようにするストリーム処理

1.3. AMQ STREAMS による KAFKA のサポート

AMQ Streams は、Kafka を OpenShift で実行するためのコンテナーイメージおよび Operator を提供します。AMQ Streams Operator は、AMQ Streams の実行に必要です。AMQ Streams で提供される Operator は、Kafka を効果的に管理するために、専門的なオペレーション情報で目的に合うよう構築されています。

Operator は以下のプロセスを単純化します。

- Kafka クラスターのデプロイおよび実行
- Kafka コンポーネントのデプロイおよび実行
- Kafka へアクセスするための設定
- Kafka へのアクセスのセキュア化
- Kafka のアップグレード
- ブローカーの管理
- トピックの作成および管理
- ユーザーの作成および管理。

1.4. OPERATOR

AMQ Streams は、OpenShift クラスター内で実行中の Kafka クラスターを管理するための Operator を提供します。

Cluster Operator

Apache Kafka クラスター、Kafka Connect、Kafka MirrorMaker、Kafka Bridge、Kafka Exporter、および Entity Operator をデプロイおよび管理します。

Entitiy Operator

Topic Operator および User Operator を設定します。

Topic Operator

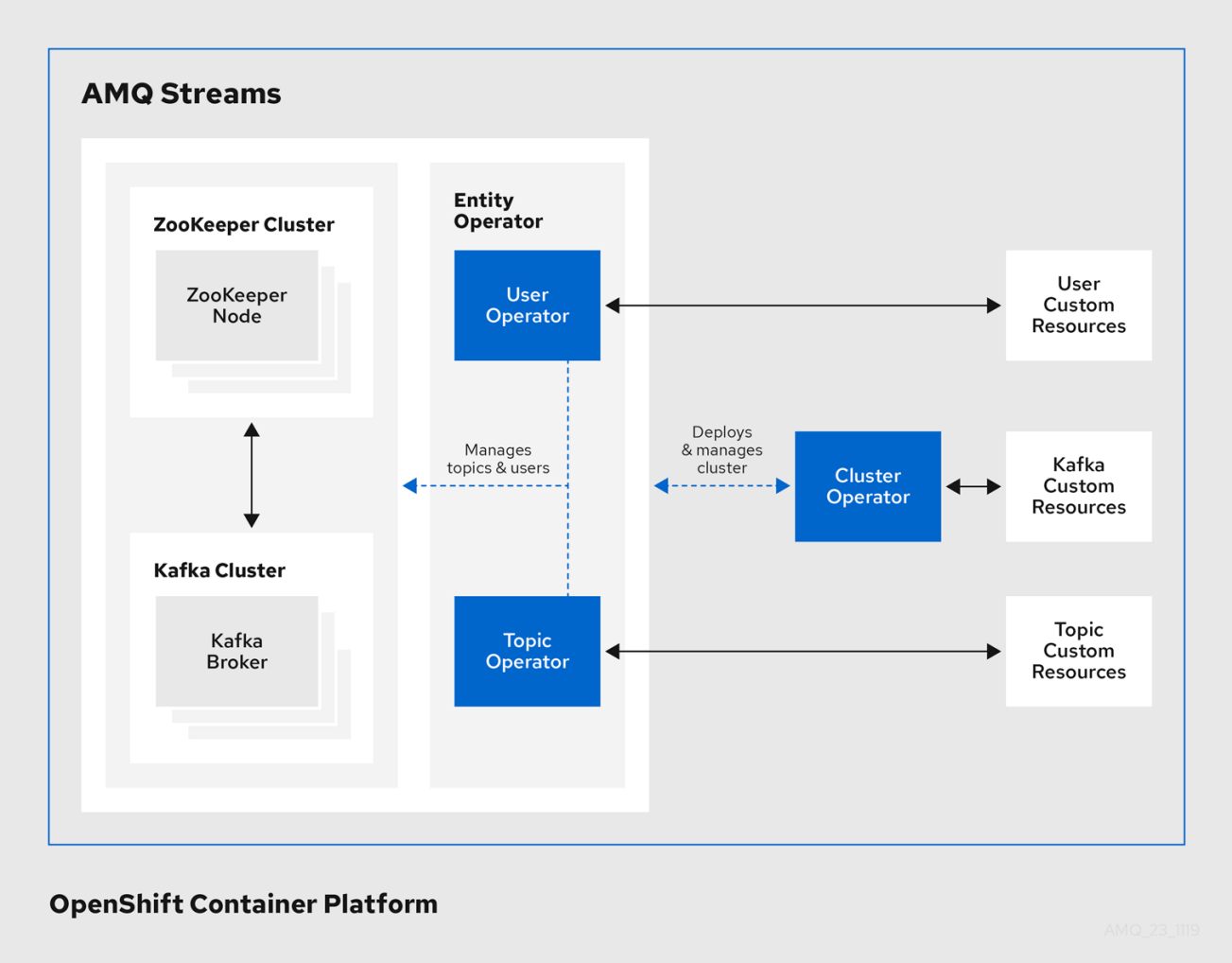
Kafka トピックを管理します。

User Operator

Kafka ユーザーを管理します。

Cluster Operator は、Kafka クラスターと同時に、Topic Operator および User Operator を **Entity Operator** 設定の一部としてデプロイできます。

AMQ Streams アーキテクチャー内の Operator



1.5. 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、**monospace** フォントのイタリック体、大文字、およびハイフン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 AMQ STREAMS の評価

本章の手順では、AMQ Streams の機能を評価する簡単な方法を説明します。

AMQ Streams をインストールし、トピックからメッセージの送受信を開始するには、指定の順序で手順にしたがいます。

- 必要な前提条件があることを確認します。
- AMQ Streams をインストールします。
- Kafka クラスターを作成します。
- Kafka クラスターにセキュアにアクセスするための認証の有効にします。
- Kafka クラスターにアクセスし、メッセージを送受信します。

前提条件を確認し、本章に指定されている順序でタスクを行うようにしてください。

2.1. 前提条件

- AMQ Streams をデプロイする OpenShift Container Platform クラスター (3.11 以上) が稼働している必要があります。
- AMQ Streams のダウンロードサイト にアクセスできる必要があります。

2.2. AMQ STREAMS のダウンロード

ZIP ファイルには、AMQ Streams のインストールに必要なリソースと、設定の例が含まれています。

手順

1. サブスクリプションがアクティベートされ、システムが登録されていることを確認します。カスタマーポータルを使用して Red Hat サブスクリプションをアクティブ化し、システムを登録する方法は、付録A *サブスクリプションの使用* を参照してください。
2. AMQ Streams のダウンロードサイト から **amq-streams-x.y.z-ocp-install-examples.zip** ファイルをダウンロードします。
3. ファイルを任意の場所で展開します。
   - Windows または Mac の場合: ZIP ファイルをダブルクリックして ZIP アーカイブの内容を展開します。
   - Red Hat Enterprise Linux の場合: ターゲットマシンでターミナルウィンドウを開き、ZIP ファイルがダウンロードされた場所に移動します。

   以下のコマンドを使用して、ZIP ファイルを展開します。

   ```
   unzip amq-streams-x.y.z-ocp-install-examples.zip
   ```

2.3. AMQ STREAMS のインストール

AMQ Streams をデプロイメントに必要な CRD (Custom Resource Definition) でインストールします。

このタスクでは、デプロイメント用にクラスターで namespace を作成します。namespace を使用して関数を分離することが推奨されます。

前提条件

- インストールには、**system:admin** などの **cluster-admin** ロールを持つユーザーが必要です。

手順

1. クラスター管理者権限を持つアカウントを使用して OpenShift クラスターにログインします。以下に例を示します。

```
oc login -u system:admin
```

2. AMQ Streams Kafka Cluster Operator に新規の **kafka** (プロジェクト) namespace を作成します。

```
oc new-project kafka
```

3. AMQ Streams Kafka Cluster Operator をインストールする新しい **kafka** namespace を参照するインストールファイルを変更します。

> **注記**
>
> デフォルトでは、ファイルは **myproject** namespace で動作します。

- Linux の場合は、以下を使用します。

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

- Mac の場合は、以下を使用します。

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

4. CRD を管理するために、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。

```
oc project kafka
oc apply -f install/cluster-operator/
```

5. Kafka クラスターをデプロイする新しい **my-kafka-project** namespace を作成します。

```
oc new-project my-kafka-project
```

6. 管理者でないユーザー **developer** に **my-kafka-project** へのアクセス権限を付与します。以下に例を示します。

```
oc adm policy add-role-to-user admin developer -n my-kafka-project
```

7. **STRIMZI_NAMESPACE** 環境変数の値を設定し、**my-kafka-project** namespace を監視する権限を Cluster Operator に付与します。

–

```
oc set env deploy/strimzi-cluster-operator STRIMZI_NAMESPACE=kafka,my-kafka-project -
n kafka
```

```
oc apply -f install/cluster-operator/020-RoleBinding-strimzi-cluster-operator.yaml -n my-kafka-
project
```

```
oc apply -f install/cluster-operator/032-RoleBinding-strimzi-cluster-operator-topic-operator-
delegation.yaml -n my-kafka-project
```

```
oc apply -f install/cluster-operator/031-RoleBinding-strimzi-cluster-operator-entity-operator-
delegation.yaml -n my-kafka-project
```

このコマンドによって、Cluster Operator が Kafka クラスターにアクセスするための権限を付与するロールバインディングが作成されます。

8. 新しいクラスターロール **strimzi-admin** を作成します。

```
oc apply -f install/strimzi-admin
```

9. ロールを管理者でないユーザー **developer** に追加します。

```
oc adm policy add-cluster-role-to-user strimzi-admin developer
```

2.4. クラスターの作成

AMQ Streams がインストールされている状態で、Kafka クラスターを作成し、クラスター内でトピックを作成します。

クラスターの作成時に、AMQ Streams のインストール時にデプロイされた Cluster Operator によって新規の Kafka リソースが監視されます。

前提条件

- Kafka クラスターでは、Cluster Operator がデプロイされている必要があります。
- トピックでは、Kafka クラスターが稼働中である必要があります。

手順

1. ユーザー **developer** として **my-kafka-project** namespace にログインします。
以下に例を示します。

```
oc login -u developer
oc project my-kafka-project
```

新規ユーザーが OpenShift Container Platform にログインした後に、そのユーザーのアカウントが作成されます。

2. 3 つの Zookeeper ノードと 3 つのブローカーノードで、新しい **my-cluster** Kafka クラスターを作成します。

- **ephemeral** ストレージの使用

- **route** を使用するよう設定された外部リスナーを使用して、OpenShift クラスター外部の Kafka クラスターを公開します。

```
cat << EOF | oc create -f -
apiVersion: kafka.strimzi.io/v1beta1
kind: Kafka
metadata:
  name: my-cluster
spec:
  kafka:
    replicas: 3
    listeners:
      plain: {}
      tls: {}
      external:
        type: route
    storage:
      type: ephemeral
  zookeeper:
    replicas: 3
    storage:
      type: ephemeral
  entityOperator:
    topicOperator: {}
EOF
```

3. クラスターがデプロイされるまで待機します。

```
oc wait my-kafka-project/my-cluster --for=condition=Ready --timeout=300s -n kafka
```

4. クラスターの準備ができたら、パブリッシュするトピックを作成し、外部クライアントからサブスクライブします。
my-cluster Kafka クラスターで 3 つのレプリカと 3 つのパーティションを使用して、以下の **my-topic** カスタムリソース定義を作成します。

```
cat << EOF | oc create -f -
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaTopic
metadata:
  name: my-topic
  labels:
    strimzi.io/cluster: "my-cluster"
spec:
  partitions: 3
  replicas: 3
EOF
```

2.5. クラスターへのアクセス

route がクラスターへの外部アクセスに使用される場合、ブローカーとクライアント間で TLS (Transport Layer Security) 暗号化を有効にするためにクラスターの CA 証明書が必要になります。

前提条件

- OpenShift クラスター内で Kafka クラスターが稼働中である必要があります。
- Cluster Operator も稼働している必要があります。

手順

1. ブートストラップ **route** のアドレスを検索します。

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

このアドレスと Kafka クライアントの 443 番ポートをブートストラップアドレスとして使用します。

2. ブローカーの認証局の公開証明書を抽出します。

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

3. 信頼できる証明書をトラストストアにインポートします。

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

メッセージの送受信を開始する準備が整いました。

2.6. トピックからのメッセージの送受信

クラスター外部のメッセージを **my-topic** から送受信すると、AMQ Streams インストールをテストできます。

ターミナルを使用して、ローカルマシンで Kafka プロデューサーおよびコンシューマーを実行します。

前提条件

- AMQ Streams が OpenShift クラスターにインストールされていることを確認します。
- メッセージを送受信するには、Zookeeper および Kafka が稼働している必要があります。
- クラスターにアクセスするためにクラスター CA 証明書 が必要です。
- AMQ Streams のダウンロードサイト から最新バージョンの Red Hat AMQ Stremas アーカイブにアクセスできる必要があります。

手順

1. AMQ Streams のダウンロードサイト から最新バージョンの AMQ Stremas アーカイブ (**amq-streams-x.y.z-bin.zip**) をダウンロードします。
ファイルを任意の場所で展開します。

2. ターミナルを開き、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。

```
bin/kafka-console-producer.sh --broker-list ROUTE-ADDRESS: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ンソール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. 新しいターミナル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。

```
bin/kafka-console-consumer.sh --bootstrap-server ROUTE-ADDRESS: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。

付録A サブスクリプションの使用

AMQ Streams 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. **JBOSS INTEGRATION AND AUTOMATION** カテゴリーの **Red Hat AMQ Streams** エントリーを見つけます。
3. 必要な AMQ Streams 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 **ダウンロード** リンクをクリックします。

改訂日時：2023-01-28 12:18:20 +1000