Evaluating AMQ Streams on OpenShift

For use with AMQ Streams 1.3 on OpenShift Container Platform
Abstract

This guide describes how to install and manage AMQ Streams to evaluate its potential use in a production environment.
# Table of Contents

## CHAPTER 1. OVERVIEW OF AMQ STREAMS
- 1.1. KAFKA KEY FEATURES .................................................. 3
- 1.2. DOCUMENT CONVENTIONS ........................................... 4

## CHAPTER 2. TRY AMQ STREAMS
- 2.1. PREREQUISITES .......................................................... 5
- 2.2. DOWNLOADING AMQ STREAMS ..................................... 5
- 2.3. INSTALLING AMQ STREAMS .......................................... 5
- 2.4. CREATING A CLUSTER ............................................... 7
- 2.5. ACCESSING THE CLUSTER .......................................... 8
- 2.6. SENDING AND RECEIVING MESSAGES FROM A TOPIC .... 9

## APPENDIX A. USING YOUR SUBSCRIPTION
- Accessing Your Account ..................................................... 10
- Activating a Subscription ................................................... 10
- Downloading Zip and Tar Files .......................................... 10
CHAPTER 1. OVERVIEW OF AMQ STREAMS

AMQ Streams is based on Apache Kafka, a popular platform for streaming data delivery and processing. AMQ Streams makes it easy to run Apache Kafka on OpenShift.

AMQ Streams provides three operators:

**Cluster Operator**
- Responsible for deploying and managing Apache Kafka clusters within an OpenShift cluster.

**Topic Operator**
- Responsible for managing Kafka topics within a Kafka cluster running within an OpenShift cluster.

**User Operator**
- Responsible for managing Kafka users within a Kafka cluster running within an OpenShift cluster.

**NOTE**

The Cluster Operator can deploy the Topic Operator and User Operator (as part of an Entity Operator configuration) at the same time as a Kafka cluster.

Operators within the AMQ Streams architecture

1.1. KAFKA KEY FEATURES
- Designed for horizontal scalability
- Message ordering guarantee at the partition level
- Message rewind/replay
  - "Long term" storage allows the reconstruction of an application state by replaying the messages
  - Combines with compacted topics to use Kafka as a key-value store

**Additional resources**

- For more information about Apache Kafka, see the [Apache Kafka website](https://kafka.apache.org).

## 1.2. DOCUMENT CONVENTIONS

### Replaceables

In this document, replaceable text is styled in monospace and italics.

For example, in the following code, you will want to replace `my-namespace` with the name of your namespace:

```bash
sed -i 's/namespace: .*/namespace: my-namespace/' install/cluster-operator/*RoleBinding*.yaml
```
CHAPTER 2. TRY AMQ STREAMS

Install AMQ Streams and start sending and receiving messages from a topic in minutes.

You will:

- Install AMQ Streams
- Create a Kafka cluster
- Access the Kafka cluster to send and receive messages

Ensure you have the prerequisites and then follow the tasks in the order provided in this chapter.

2.1. PREREQUISITES

- OpenShift Container Platform cluster (3.11 and later) running on which to deploy AMQ Streams

2.2. DOWNLOADING AMQ STREAMS

Download a zip file that contains the resources required for installation along with examples for configuration.

Prerequisites

- Access to the AMQ Streams download site.

Procedure

1. Download the `amq-streams-x.y.z-ocp-install-examples.zip` file from the AMQ Streams download site.

2. Unzip the file to any destination.

   - On Windows or Mac, you can extract the contents of the ZIP archive by double clicking on the ZIP file.

   - On Red Hat Enterprise Linux, open a terminal window in the target machine and navigate to where the ZIP file was downloaded. Extract the ZIP file by executing the following command:

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

2.3. INSTALLING AMQ STREAMS

Install AMQ Streams with the Custom Resource Definitions (CRDs) required for deployment.

In this task you create namespaces in the cluster for your deployment. It is good practice to use namespaces to separate functions.

Prerequisites

- Installation requires a user with `cluster-admin` role, such as `system:admin`
Procedure

1. Login in to the OpenShift cluster with cluster admin privileges.
   For example:
   ```
   oc login -u system:admin
   ```
2. Create a new kafka (project) namespace for the AMQ Streams Kafka Cluster Operator.
   ```
   oc new-project kafka
   ```
3. Modify the installation files to reference the new kafka namespace where you will install the AMQ Streams Kafka Cluster Operator.
   **NOTE**
   By default, the files work in the myproject namespace.
   ```
   • On Linux, use:
     ```
     sed -i 's/namespace: .*/namespace: kafka/' install/cluster-operator/*RoleBinding*.yaml
     ```
   • On Mac, use:
     ```
     sed -i '' 's/namespace: .*/namespace: kafka/' install/cluster-operator/*RoleBinding*.yaml
     ```
   ```

4. Deploy the CRDs and role-based access control (RBAC) resources to manage the CRDs.
   ```
   oc new-project kafka
   oc apply -f install/cluster-operator/
   ```
5. Create a new my-kafka-project namespace where you will deploy your Kafka cluster.
   ```
   oc new-project my-kafka-project
   ```
6. Give access to my-kafka-project to a non-admin user developer.
   For example:
   ```
   oc adm policy add-role-to-user admin developer -n my-kafka-project
   ```
7. Give permission to the Cluster Operator to watch the my-kafka-project namespace.
   ```
   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
   ```
The commands create role bindings that grant permission for the Cluster Operator to access the Kafka cluster.

8. Create a new cluster role `strimzi-admin`.

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

9. Add the role to the non-admin user `developer`.

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

### 2.4. CREATING A CLUSTER

Create a Kafka cluster, then a topic within the cluster.

When you create a cluster, the Cluster Operator you deployed watches for new Kafka resources.

**Prerequisites**

- For the Kafka cluster, a Cluster Operator is deployed
- For the topic, a running Kafka cluster

**Procedure**

1. Log in to the `my-kafka-project` namespace as user `developer`.
   For example:

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

   After new users log in to OpenShift Container Platform, an account is created for that user.

2. Create a new `my-cluster` Kafka cluster with 3 Zookeeper and 3 broker nodes.
   
   - Use `ephemeral` storage
   - Expose the Kafka cluster outside of the OpenShift cluster using an external listener configured to use `route`.

   ```yaml
   cat << EOF | oc create -f -
   apiVersion: kafka.strimzi.io/v1beta1
   kind: Kafka
   metadata:
     name: my-cluster
   spec:
     kafka:
       replicas: 3
       listeners:
         plain: {}
         tls: {}
   EOF
   ```
3. Wait for the cluster to be deployed:

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

4. Now that your cluster is running, create a topic to publish and subscribe from your external client.

   Create the following `my-topic` custom resource definition with 3 replicas and 3 partitions in the `my-cluster` Kafka cluster:

   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. ACCESSING THE CLUSTER

As `route` is used for external access to the cluster, a cluster CA certificate is required to enable TLS (Transport Layer Security) encryption between the broker and the client.

**Prerequisites**

- A Kafka cluster running within the OpenShift cluster
- A running Cluster Operator

**Procedure**

1. Find the address of the bootstrap `route`:

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

   Use the address together with port 443 in your Kafka client as the bootstrap address.

2. Extract the public certificate of the broker certification authority:
oc extract secret/my-cluster-cluster-ca-cert --keys=ca.crt --to=- > ca.crt

3. Import the trusted certificate to a truststore:

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

You are now ready to start sending and receiving messages.

2.6. SENDING AND RECEIVING MESSAGES FROM A TOPIC

Test your AMQ Streams installation by sending and receiving messages outside the cluster from my-topic.

Use a terminal to run a Kafka producer and consumer on a local machine.

Prerequisites

- AMQ Streams is installed on the OpenShift cluster
- Zookeeper and Kafka are running
- Cluster CA certificate for access to the cluster
- Access to the latest version of the Red Hat AMQ Streams archive from the AMQ Streams download site

Procedure

1. Download the latest version of the AMQ Streams archive (amq-streams-x.y.z-bin.zip) from the AMQ Streams download site. Unzip the file to any destination.

2. Open a terminal and start the Kafka console producer with the topic my-topic and the authentication properties for TLS:

   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. Type your message into the console where the producer is running.

4. Press Enter to send the message.

5. Open a new terminal tab or window and start the Kafka console consumer to receive the messages:

   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. Confirm that you see the incoming messages in the consumer console.

7. Press Ctrl+C to exit the Kafka console producer and consumer.
APPENDIX A. USING YOUR SUBSCRIPTION

AMQ Streams is provided through a software subscription. To manage your subscriptions, access your account at the Red Hat Customer Portal.

Accessing Your Account

1. Go to access.redhat.com.
2. If you do not already have an account, create one.
3. Log in to your account.

Activating a Subscription

1. Go to access.redhat.com.
2. Navigate to My Subscriptions.
3. Navigate to Activate a subscription and enter your 16-digit activation number.

Downloading Zip and Tar Files
To access zip or tar files, use the customer portal to find the relevant files for download. If you are using RPM packages, this step is not required.

1. Open a browser and log in to the Red Hat Customer Portal Product Downloads page at access.redhat.com/downloads.
2. Locate the Red Hat AMQ Streams entries in the JBOSS INTEGRATION AND AUTOMATION category.
3. Select the desired AMQ Streams product. The Software Downloads page opens.
4. Click the Download link for your component.

Revised on 2019-11-07 07:14:09 UTC