Red Hat AMQ Streams 2.6

Getting Started with AMQ Streams on OpenShift

Get started using AMQ Streams 2.6 on OpenShift Container Platform

Last Updated: 2023-12-06

Red Hat AMQ Streams 2.6 Getting Started with AMQ Streams on OpenShift

Get started using AMQ Streams 2.6 on OpenShift Container Platform

Legal Notice

Abstract

Try AMQ Streams by creating a Kafka cluster on OpenShift. Connect to the Kafka cluster, then send and receive messages from a Kafka topic.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

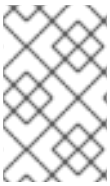
Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

CHAPTER 1. GETTING STARTED OVERVIEW

Use Red Hat AMQ Streams to create and set up Kafka clusters, then connect your applications and services to those clusters.

This guide describes how to install and start using AMQ Streams on OpenShift Container Platform. You can install the AMQ Streams operator directly from the OperatorHub in the OpenShift web console. The AMQ Streams operator understands how to install and manage Kafka components. Installing from the OperatorHub provides a standard configuration of AMQ Streams that allows you to take advantage of automatic updates.

When the AMQ Streams operator is installed, it provides the resources to install instances of Kafka components. After installing a Kafka cluster, you can start producing and consuming messages.



NOTE

If you require more flexibility with your deployment, you can use the installation artifacts provided with AMQ Streams. For more information on using the installation artifacts, see *Deploying and Managing AMQ Streams on OpenShift* .

1.1. PREREQUISITES

The following prerequisites are required for getting started with AMQ Streams.

- You have a Red Hat account.
- JDK 11 or later is installed.
- An OpenShift 4.11 to 4.14 cluster is available.
- The OpenShift **oc** command-line tool is installed and configured to connect to the running cluster.

The steps to get started are based on using the OperatorHub in the OpenShift web console, but you'll also use the OpenShift **oc** CLI tool to perform certain operations. You'll need to connect to your OpenShift cluster using the **oc** tool.

- You can install the **oc** CLI tool from the web console by clicking the **'?'** help menu, then **Command Line Tools**.
- You can copy the required **oc login** details from the web console by clicking your profile name, then **Copy login command**.

1.2. ADDITIONAL RESOURCES

- Strimzi Overview
- Deploying and Upgrading AMQ Streams on OpenShift

CHAPTER 2. INSTALLING THE AMQ STREAMS OPERATOR FROM THE OPERATORHUB

You can install and subscribe to the AMQ Streams operator using the OperatorHub in the OpenShift Container Platform web console.

This procedure describes how to create a project and install the AMQ Streams operator to that project. A project is a representation of a namespace. For manageability, it is a good practice to use namespaces to separate functions.

WARNING

Make sure you use the appropriate update channel. If you are on a supported version of OpenShift, installing AMQ Streams from the default stable channel is generally safe. However, we do not recommend enabling automatic updates on the stable channel. An automatic upgrade will skip any necessary steps prior to upgrade. Use automatic upgrades only on version-specific channels.

Prerequisites

- Access to an OpenShift Container Platform web console using an account with **cluster-admin** or **strimzi-admin** permissions.

Procedure

1. Navigate in the OpenShift web console to the **Home > Projects** page and create a project (namespace) for the installation.
   We use a project named **amq-streams-kafka** in this example.
2. Navigate to the **Operators > OperatorHub** page.
3. Scroll or type a keyword into the **Filter by keyword** box to find the **AMQ Streams** operator.
   The operator is located in the **Streaming & Messaging** category.
4. Click **AMQ Streams** to display the operator information.
5. Read the information about the operator and click **Install**.
6. On the **Install Operator** page, choose from the following installation and update options:
   - **Update Channel**: Choose the update channel for the operator.
     - The (default) **stable** channel contains all the latest updates and releases, including major, minor, and micro releases, which are assumed to be well tested and stable.
     - An **amq-streams-*X*.x** channel contains the minor and micro release updates for a major release, where *X* is the major release version number.
     - An **amq-streams-*X.Y*.x** channel contains the micro release updates for a minor release, where *X* is the major release version number and *Y* is the minor release version number.

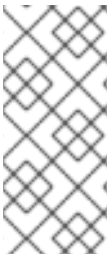
- **Installation Mode**: Choose the project you created to install the operator on a specific namespace.
  You can install the AMQ Streams operator to all namespaces in the cluster (the default option) or a specific namespace. We recommend that you dedicate a specific namespace to the Kafka cluster and other AMQ Streams components.

- **Update approval**: By default, the AMQ Streams operator is automatically upgraded to the latest AMQ Streams version by the Operator Lifecycle Manager (OLM). Optionally, select **Manual** if you want to manually approve future upgrades. For more information on operators, see the OpenShift documentation.

7. Click **Install** to install the operator to your selected namespace.
   The AMQ Streams operator deploys the Cluster Operator, CRDs, and role-based access control (RBAC) resources to the selected namespace.

8. After the operator is ready for use, navigate to **Operators > Installed Operators** to verify that the operator has installed to the selected namespace.
   The status will show as **Succeeded**.

   You can now use the AMQ Streams operator to deploy Kafka components, starting with a Kafka cluster.



NOTE

If you navigate to **Workloads > Deployments**, you can see the deployment details for the Cluster Operator and Entity Operator. The name of the Cluster Operator includes a version number: **amq-streams-cluster-operator-<version>**. The name is different when deploying the Cluster Operator using the AMQ Streams installation artifacts. In this case, the name is **strimzi-cluster-operator**.

CHAPTER 3. DEPLOYING KAFKA COMPONENTS USING THE AMQ STREAMS OPERATOR

When installed on Openshift, the AMQ Streams operator makes Kafka components available for installation from the user interface.

The following Kafka components are available for installation:

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

You select the component and create an instance. As a minimum, you create a Kafka instance. This procedure describes how to create a Kafka instance using the default settings. You can configure the default installation specification before you perform the installation.

The process is the same for creating instances of other Kafka components.

Prerequisites

- The AMQ Streams operator is installed on the OpenShift cluster.

Procedure

1. Navigate in the web console to the **Operators > Installed Operators** page and click **AMQ Streams** to display the operator details.
   From **Provided APIs**, you can create instances of Kafka components.

2. Click **Create instance** under **Kafka** to create a Kafka instance.
   By default, you'll create a Kafka cluster called **my-cluster** with three Kafka broker nodes and three ZooKeeper nodes. The cluster uses ephemeral storage.

3. Click **Create** to start the installation of Kafka.
   Wait until the status changes to **Ready**.

CHAPTER 4. CREATING AN OPENSHIFT ROUTE TO ACCESS A KAFKA CLUSTER

Create an OpenShift route to access a Kafka cluster outside of OpenShift.

This procedure describes how to expose a Kafka cluster to clients outside the OpenShift environment. After the Kafka cluster is exposed, external clients can produce and consume messages from the Kafka cluster.

To create an OpenShift route, a **route** listener is added to the configuration of a Kafka cluster installed on OpenShift.

WARNING

An OpenShift Route address includes the name of the Kafka cluster, the name of the listener, and the name of the namespace it is created in. For example, **my-cluster-kafka-listener1-bootstrap-amq-streams-kafka** (*<cluster_name>*-kafka-*<listener_name>*-bootstrap-*<namespace>*). Be careful that the whole length of the address does not exceed a maximum limit of 63 characters.

Prerequisites

- You have created a Kafka cluster on OpenShift .
- You need the OpenJDK **keytool** to manage certificates.
- (Optional) You can perform some of the steps using the OpenShift **oc** CLI tool.

Procedure

1. Navigate in the web console to the **Operators > Installed Operators** page and select **AMQ Streams** to display the operator details.
2. Select the **Kafka** page to show the installed Kafka clusters.
3. Click the name of the Kafka cluster you are configuring to view its details.
   We use a Kafka cluster named **my-cluster** in this example.
4. Select the **YAML** page for the Kafka cluster **my-cluster**.
5. Add route listener configuration to create an OpenShift route named **listener1**.
   The listener configuration must be set to the **route** type. You add the listener configuration under **listeners** in the Kafka configuration.

   External route listener configuration

```
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
# ...
```

The client connects on port 443, the default router port, but traffic is then routed to the port you configure, which is 9094 in this example.

6. Save the updated configuration.

7. Select the **Resources** page for the Kafka cluster **my-cluster** to locate the connection information you will need for your client.
   From the **Resources** page, you'll find details for the route listener and the public cluster certificate you need to connect to the Kafka cluster.

8. Click the name of the **my-cluster-kafka-listener1-bootstrap** route created for the Kafka cluster to show the route details.

9. Make a note of the hostname.
   The hostname is specified with port 443 in a Kafka client as the bootstrap address for connecting to the Kafka cluster.

   You can also locate the bootstrap address by navigating to **Networking > Routes** and selecting the **amq-streams-kafka** project to display the routes created in the namespace.

   Or you can use the **oc** tool to extract the bootstrap details.

   Extracting bootstrap information

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

10. Navigate back to the **Resources** page and click the name of the **my-cluster-cluster-ca-cert** to show the secret details for accessing the Kafka cluster.
    The **ca.crt** certificate file contains the public certificate of the Kafka cluster.

    You will need the certificate to access the Kafka broker.

11. Make a local copy of the **ca.crt** public certificate file.
    You can copy the details of the certificate or use the OpenShift **oc** tool to extract them.

    Extracting the public certificate

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

12. Create a local truststore for the public cluster certificate using **keytool**.

    Creating a local truststore

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

When prompted, create a password for accessing the truststore.

The truststore is specified in a Kafka client for authenticating access to the Kafka cluster.

You are now ready to start sending and receiving messages.

CHAPTER 5. SENDING AND RECEIVING MESSAGES FROM A TOPIC

Send messages to and receive messages from a Kafka cluster installed on OpenShift.

This procedure describes how to use Kafka clients to produce and consume messages. You can deploy clients to OpenShift or connect local Kafka clients to the OpenShift cluster. You can use either or both options to test your Kafka cluster installation. For the local clients, you access the Kafka cluster using an OpenShift route connection.

You will use the **oc** command-line tool to deploy and run the Kafka clients.

Prerequisites

- You have created a Kafka cluster on OpenShift .

For a local producer and consumer:

- You have created a route for external access to the Kafka cluster running in OpenShift .
- You can access the latest Kafka client binaries from the AMQ Streams software downloads page.

Sending and receiving messages from Kafka clients deployed to the OpenShift cluster

Deploy producer and consumer clients to the OpenShift cluster. You can then use the clients to send and receive messages from the Kafka cluster in the same namespace. The deployment uses the AMQ Streams container image for running Kafka.

1. Use the **oc** command-line interface to deploy a Kafka producer.
   This example deploys a Kafka producer that connects to the Kafka cluster **my-cluster**

   A topic named **my-topic** is created.

   Deploying a Kafka producer to OpenShift

   ```
   oc run kafka-producer -ti \
   --image=registry.redhat.io/amq-streams/kafka-36-rhel8:2.6.0 \
   --rm=true \
   --restart=Never \
   -- bin/kafka-console-producer.sh \
   --bootstrap-server my-cluster-kafka-bootstrap:9092 \
   --topic my-topic
   ```

   NOTE

   If the connection fails, check that the Kafka cluster is running and the correct cluster name is specified as the **bootstrap-server**.

2. From the command prompt, enter a number of messages.
3. Navigate in the OpenShift web console to the **Home > Projects** page and select the **amq-streams-kafka** project you created.
4. From the list of pods, click **kafka-producer** to view the producer pod details.

5. Select **Logs** page to check the messages you entered are present.

6. Use the **oc** command-line interface to deploy a Kafka consumer.

   Deploying a Kafka consumer to OpenShift

   ```
   oc run kafka-consumer -ti \
   --image=registry.redhat.io/amq-streams/kafka-36-rhel8:2.6.0 \
   --rm=true \
   --restart=Never \
   -- bin/kafka-console-consumer.sh \
   --bootstrap-server my-cluster-kafka-bootstrap:9092 \
   --topic my-topic \
   --from-beginning
   ```

   The consumer consumed messages produced to **my-topic**.

7. From the command prompt, confirm that you see the incoming messages in the consumer console.

8. Navigate in the OpenShift web console to the **Home > Projects** page and select the **amq-streams-kafka** project you created.

9. From the list of pods, click **kafka-consumer** to view the consumer pod details.

10. Select the **Logs** page to check the messages you consumed are present.

Sending and receiving messages from Kafka clients running locally

Use a command-line interface to run a Kafka producer and consumer on a local machine.

1. Download and extract the **AMQ Streams *<version>*** binaries from the AMQ Streams software downloads page.
   Unzip the **amq-streams-*<version>*-bin.zip** file to any destination.

2. Open a command-line interface, and start the Kafka console producer with the topic **my-topic** and the authentication properties for TLS.
   Add the properties that are required for accessing the Kafka broker with an OpenShift route .

   - Use the hostname and port 443 for the OpenShift route you are using.
   - Use the password and reference to the truststore you created for the broker certificate.

   Starting a local Kafka producer

   ```
   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. Type your message into the command-line interface where the producer is running.

4. Press enter to send the message.

5. Open a new command-line interface tab or window, and start the Kafka console consumer to receive the messages.
   Use the same connection details as the producer.

   Starting a local Kafka consumer

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

7. Press Crtl+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 **AMQ Streams for Apache Kafka** entries in the **INTEGRATION AND AUTOMATION** category.
3. Select the desired AMQ Streams product. The **Software Downloads** page opens.
4. Click the **Download** link for your component.

Installing packages with DNF

To install a package and all the package dependencies, use:

```
dnf install <package_name>
```

To install a previously-downloaded package from a local directory, use:

```
dnf install <path_to_download_package>
```

Revised on 2023-12-06 17:40:03 UTC