



Red Hat Streams for Apache Kafka 2.7

在 OpenShift 中使用 Apache Kafka 的流

在 OpenShift Container Platform 上开始使用 Apache Kafka 2.7 的 Streams

Red Hat Streams for Apache Kafka 2.7 在 OpenShift 中使用 Apache Kafka 的流

在 OpenShift Container Platform 上开始使用 Apache Kafka 2.7 的 Streams

法律通告

Copyright © 2024 Red Hat, Inc.

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.

摘要

通过在 OpenShift 中创建 Kafka 集群来尝试 Apache Kafka 的流。连接到 Kafka 集群，然后从 Kafka 主题发送和接收信息。

目录

前言	3
对红帽文档提供反馈	4
第 1 章 入门概述	5
1.1. 先决条件	5
1.2. 其他资源	5
第 2 章 从 OPERATORHUB 安装 APACHE KAFKA OPERATOR 的 STREAMS	6
第 3 章 使用 APACHE KAFKA OPERATOR 的 STREAMS 部署 KAFKA 组件	8
第 4 章 创建 OPENSIFT 路由以访问 KAFKA 集群	9
第 5 章 从主题发送和接收信息	11
第 6 章 (预览) 为 APACHE KAFKA 控制台部署流	14
附录 A. 使用您的订阅	15
访问您的帐户	15
激活订阅	15
下载 Zip 和 Tar 文件	15
使用 DNF 安装软件包	15

前言

对红帽文档提供反馈

我们感谢您对我们文档的反馈。

要改进，创建一个 JIRA 问题并描述您推荐的更改。提供尽可能多的详细信息，以便我们快速解决您的请求。

前提条件

- 您有红帽客户门户网站帐户。此帐户可让您登录到 Red Hat Jira Software 实例。如果您没有帐户，系统会提示您创建一个帐户。

流程

1. 点以下内容：[Create issue](#)。
2. 在 **Summary** 文本框中输入问题的简短描述。
3. 在 **Description** 文本框中提供以下信息：
 - 找到此问题的页面的 URL。
 - 有关此问题的详细描述。
您可以将信息保留在任何其他字段中的默认值。
4. 添加 reporter 名称。
5. 点 **Create** 将 JIRA 问题提交到文档团队。

感谢您花时间来提供反馈。

第 1 章 入门概述

使用 Red Hat Streams for Apache Kafka 创建和设置 Kafka 集群，然后将您的应用程序和服务连接到这些集群。

本指南论述了如何在 OpenShift Container Platform 上安装和启动 Apache Kafka 的 Streams。您可以直接从 OpenShift Web 控制台中的 OperatorHub 安装 Apache Kafka operator 的 Streams。Apache Kafka operator 的 Streams 了解如何安装和管理 Kafka 组件。从 OperatorHub 安装为 Apache Kafka 提供了标准配置 Apache Kafka，可让您利用自动更新。

安装 Apache Kafka operator 的 Streams 时，它提供安装 Kafka 组件实例的资源。安装 Kafka 集群后，您可以启动生成和使用消息。



注意

如果部署需要更大的灵活性，您可以使用 Apache Kafka 提供的 Streams 提供的安装工件。有关使用安装工件的更多信息，请参阅在 [OpenShift 中部署和管理 Apache Kafka 的流](#)。

1.1. 先决条件

使用 Apache Kafka 的 Streams 开始需要满足以下先决条件。

- 您有红帽帐户。
- 已安装了 JDK 11 或更高版本。
- OpenShift 4.12 到 4.15 集群可用。
- OpenShift **oc** 命令行工具已安装并配置为连接到正在运行的集群。

入门步骤基于 OpenShift Web 控制台中的 OperatorHub，但您也可以使用 OpenShift **oc** CLI 工具执行某些操作。您需要使用 **oc** 工具连接到 OpenShift 集群。

- 您可以通过点击 '?' help 菜单，然后点 **Command Line Tools** 来安装 **oc** CLI 工具。
- 您可以通过点击您的配置集名称，然后从 web 控制台复制所需的 **oc login** 详情，然后 **复制登录命令**。

1.2. 其他资源

- [Strimzi 概述](#)
- [在 OpenShift 中部署和升级 Apache Kafka 的流](#)

第 2 章 从 OPERATORHUB 安装 APACHE KAFKA OPERATOR 的 STREAMS

您可以使用 OpenShift Container Platform Web 控制台中的 OperatorHub 安装并订阅 Apache Kafka operator 的 Streams。

此流程描述了如何创建项目，并将 Apache Kafka operator 的 Streams 安装到该项目中。项目命名空间的表示。对于可管理性，最好使用命名空间来分隔功能。



警告

确保使用正确的更新频道。如果您在受支持的 OpenShift 版本中，从默认 stable 频道安装 Apache Kafka 的 Streams 通常是安全的。但是，我们不推荐在 stable 频道中启用自动更新。自动升级将在升级前跳过所有必要的步骤。仅在特定于版本的频道中使用自动升级。

先决条件

- 使用具有 **cluster-admin** 或 **strimzi-admin** 权限的账户访问 OpenShift Container Platform Web 控制台。

流程

1. 在 OpenShift Web 控制台中进入到 **Home > Projects** 页面，再创建一个用于安装的项目（命名空间）。
在这个示例中，我们使用名为 **amq-streams-kafka** 的项目。
2. 进入 **Operators > OperatorHub** 页面。
3. 在 **Filter by keyword** 框中滚动或输入关键字以查找 **Apache Kafka operator 的 Streams**，operator 位于 **Streaming 和 Messaging** 目录中。
4. 点 **Streams for Apache Kafka** 来显示 Operator 信息。
5. 阅读有关 Operator 的信息，再点 **Install**。
6. 在 **Install Operator** 页面中，从以下安装和更新选项中选择：
 - **更新频道**：选择 Operator 的更新频道。
 - （默认） **stable** 频道包含所有最新的更新和发行版本，包括主版本、次版本和微版本，这些版本被认为经过充分测试和稳定。
 - **amq-streams-X.x** 频道包含主发行版本的次要和微版本更新，其中 X 是主版本的版本号。
 - **amq-streams-X.Y.x** 频道包含次要发行本版本的微版本更新，其中 X 是主版本的版本号，Y 是次版本号。
 - **Installation Mode**：选择您创建的项目，以便在特定命名空间中安装 Operator。

您可以将 Apache Kafka operator 的 Streams 安装到集群中的所有命名空间（默认选项）或特定命名空间中。我们建议您将特定命名空间专用于 Kafka 集群和其他 Streams for Apache Kafka 组件。

- **更新批准**：默认情况下，Apache Kafka operator 的 Streams 由 Operator Lifecycle Manager (OLM) 自动升级到 Apache Kafka 版本的最新流。另外，如果您希望手动批准将来的升级，请选择 **Manual**。如需有关 operator 的更多信息，请参阅 [OpenShift 文档](#)。

7. 点 **Install** 将 Operator 安装到所选命名空间中。

Apache Kafka operator 的 Streams 将 Cluster Operator、CRD 和基于角色的访问控制(RBAC)资源部署到所选命名空间中。

8. Operator 就绪可用后，进入 **Operators > Installed Operators** 来验证 Operator 是否已安装到所选命名空间中。

状态将显示为 **Succeeded**。

现在，您可以使用 Apache Kafka operator 的 Streams 来部署 Kafka 组件，从 Kafka 集群开始。



注意

如果您进入到 **Workloads > Deployments**，您可以查看 Cluster Operator 和 Entity Operator 的部署详情。Cluster Operator 的名称包含一个版本号：**amq-streams-cluster-operator-<version>**。当使用 Streams for Apache Kafka 安装工件部署 Cluster Operator 时，名称会有所不同。在本例中，名称是 **strimzi-cluster-operator**。

第 3 章 使用 APACHE KAFKA OPERATOR 的 STREAMS 部署 KAFKA 组件

在 OpenShift 上安装时，Apache Kafka operator 的 Streams 使 Kafka 组件可从用户界面安装。

以下 Kafka 组件可用于安装：

- Kafka
- Kafka Connect
- Kafka MirrorMaker
- Kafka MirrorMaker 2
- Kafka 主题
- Kafka 用户
- Kafka Bridge
- Kafka Connector
- Kafka Rebalance

您可以选择组件并创建实例。您至少创建一个 Kafka 实例。这个步骤描述了如何使用默认设置创建 Kafka 实例。您可以在执行安装前配置默认安装规格。

创建其他 Kafka 组件实例的过程相同。

先决条件

- Apache Kafka operator 的 Streams [安装在 OpenShift 集群中](#)。

流程

1. 在 Web 控制台中进入 Operators > Installed Operators 页面，点 Streams for Apache Kafka 来显示 Operator 详情。
在 Provided APIs 中，您可以创建 Kafka 组件的实例。
2. 点 Kafka 下的 Create instance 创建 Kafka 实例。
默认情况下，您将创建一个名为 **my-cluster** 的 Kafka 集群，它有三个 Kafka 代理节点和三个 ZooKeeper 节点。集群使用临时存储。
3. 点 Create 开始安装 Kafka。
等待状态变为 Ready。

第 4 章 创建 OPENSIFT 路由以访问 KAFKA 集群

创建 OpenShift 路由以访问 OpenShift 之外的 Kafka 集群。

此流程描述了如何在 OpenShift 环境外将 Kafka 集群公开给客户端。在 Kafka 集群公开后，外部客户端可以从 Kafka 集群生成和使用信息。

要创建 OpenShift 路由，**route** 监听程序添加到 OpenShift 上安装的 Kafka 集群的配置中。



警告

OpenShift Route 地址包括 Kafka 集群的名称、侦听器的名称以及它在其中创建的命名空间的名称。例如，**my-cluster-kafka-listener1-bootstrap-amq-streams-kafka** (`<cluster_name>-kafka-<listener_name>-bootstrap-<namespace>`)。请注意，地址的整个长度不超过 63 个字符的最大值。

先决条件

- 您已在 [OpenShift 上创建 Kafka 集群](#)。
- 您需要 OpenJDK **keytool** 管理证书。
- (可选) 您可以使用 OpenShift **oc** CLI 工具执行一些步骤。

流程

1. 在 Web 控制台中进入 Operators > Installed Operators 页面，然后选择 **Streams for Apache Kafka** 来显示 Operator 详情。
2. 选择 **Kafka** 页面来显示已安装的 Kafka 集群。
3. 点您要配置的 Kafka 集群的名称查看其详情。
在这个示例中，我们使用名为 **my-cluster** 的 Kafka 集群。
4. 选择 Kafka 集群的 **my-cluster** 的 YAML 页面。
5. 添加路由侦听器配置，以创建名为 **listener1** 的 OpenShift 路由。
侦听器配置必须设置为 **route** 类型。您可以在 Kafka 配置中的 **listeners** 中添加监听程序配置。

外部路由监听程序配置

```
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
# ...
```

客户端通过端口 443 连接默认的路由器端口，但流量会被路由到您配置的端口，本例中为 9094。

- 保存更新的配置。
- 选择 Kafka 集群 **my-cluster** 的 **Resources** 页面，以查找客户端需要的连接信息。
在 **Resources** 页面中，您将查找路由监听程序和您连接到 Kafka 集群所需的公共集群证书的详情。
- 点为 Kafka 集群创建的 **my-cluster-kafka-listener1-bootstrap** 路由名称来显示路由详情。
- 记录主机名。
主机名在 Kafka 客户端中使用端口 443 指定，作为连接到 Kafka 集群的 bootstrap 地址。

您还可以通过进入到 **Networking > Routes** 并选择 **amq-streams-kafka** 项目来找到 bootstrap 地址。

或者，您可以使用 **oc** 工具提取 bootstrap 详情。

提取 bootstrap 信息

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

- 返回到 **Resources** 页面，然后点 **my-cluster-cluster-ca-cert** 的名称以显示用于访问 Kafka 集群的 secret 详情。
ca.crt 证书文件包含 Kafka 集群的公共证书。

您需要证书才能访问 Kafka 代理。

- 生成 **ca.crt** 公共证书文件的本地副本。
您可以复制证书的详细信息，或使用 OpenShift **oc** 工具提取证书。

提取公共证书

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

- 使用 **keytool** 为公共集群证书创建本地信任存储。

创建本地信任存储

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

出现提示时，创建用于访问 truststore 的密码。

truststore 在 Kafka 客户端中指定，用于身份验证对 Kafka 集群的访问。

现在，您已准备好开始发送和接收信息。

第 5 章 从主题发送和接收信息

向在 OpenShift 上安装的 Kafka 集群发送或接收信息。

此流程描述了如何使用 Kafka 客户端生成和使用信息。您可以将客户端部署到 OpenShift，或将本地 Kafka 客户端连接到 OpenShift 集群。您可以使用其中一个或这两个选项来测试 Kafka 集群安装。对于本地客户端，您可以使用 OpenShift 路由连接访问 Kafka 集群。

您将使用 **oc** 命令行工具来部署并运行 Kafka 客户端。

先决条件

- 您已在 [OpenShift 上创建 Kafka 集群](#)。

对于生成者和消费者：

- 您已创建了路由，供外部访问在 OpenShift 中运行的 Kafka 集群。
- 您可以从 [Apache Kafka 软件下载页面的 Streams](#) 访问最新的 Kafka 客户端二进制文件。

从部署到 OpenShift 集群的 Kafka 客户端发送和接收信息

将生成者和消费者客户端部署到 OpenShift 集群。然后，您可以使用客户端从同一命名空间中的 Kafka 集群发送和接收信息。部署使用 Streams for Apache Kafka 容器镜像来运行 Kafka。

1. 使用 **oc** 命令行界面部署 Kafka producer。
本例部署一个 Kafka 生成者，它连接到 Kafka 集群 **my-cluster**

名为 **my-topic** 的主题被创建。

将 Kafka producer 部署到 OpenShift

```
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
```



注意

如果连接失败，请检查 Kafka 集群是否正在运行，并将正确的集群名称指定为 **bootstrap-server**。

2. 从命令提示符输入多个信息。
3. 在 OpenShift web 控制台中进入到 **Home > Projects** 页面，然后选择 **amq-streams-kafka** 项目。
4. 从 pod 列表中，点 **kafka-producer** 以查看 producer pod 详情。
5. 选择 **Logs** 页面，以检查您输入的消息是否存在。
6. 使用 **oc** 命令行界面部署 Kafka 用户。

将 Kafka 使用者部署到 OpenShift

```
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
```

消费者使用生成给 **my-topic** 的消息。

7. 从命令提示符，确认您在使用者控制台中看到传入的信息。
8. 在 OpenShift web 控制台中进入到 **Home > Projects** 页面，然后选择 **amq-streams-kafka** 项目。
9. 从 pod 列表中，点 **kafka-consumer** 查看使用者 pod 详情。
10. 选择 **Logs** 页面，以检查您消耗的消息是否存在。

从本地运行的 Kafka 客户端发送和接收信息

使用命令行界面在本地机器上运行 Kafka 制作者和使用者。

1. 从 **Streams for Apache Kafka 软件下载页面** 下载并提取 **Apache Kafka <version>** 二进制文件的 **Streams**。
解压 **amq-streams-<version>-bin.zip** 文件到任何目标。
2. 打开命令行界面，并使用主题 **my-topic** 和 TLS 的身份验证属性启动 Kafka 控制台制作者。
添加使用 **OpenShift 路由访问 Kafka 代理** 所需的属性。
 - 将主机名和端口 443 用于您正在使用的 OpenShift 路由。
 - 使用密码并引用您为代理证书创建的信任存储。

启动本地 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. 在运行制作者的命令行界面中键入您的消息。
4. 按 **enter** 来发送邮件。
5. 打开新的命令行界面标签页或窗口，并启动 Kafka 控制台使用者来接收信息。
使用与生成者相同的连接详情。

启动本地 Kafka 消费者

```
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. 确认您在使用者控制台中看到传入的信息。
7. 按 Ctrl+C 退出 Kafka 控制台生成者和消费者。

第 6 章（预览）为 APACHE KAFKA 控制台部署流

部署由 Streams for Apache Kafka 管理的 Kafka 集群后，您可以部署 Apache Kafka 控制台的 Streams 并连接集群。Apache Kafka 控制台的 Streams 有助于管理 Kafka 集群，通过其用户界面为监控、管理和优化每个集群提供实时见解。

有关连接到 Apache Kafka 控制台的流以及使用 Apache Kafka 控制台的更多信息，[请参阅 Apache Kafka 文档中的 控制台指南](#)。



注意

Apache Kafka 控制台的流当前作为技术预览提供。

附录 A. 使用您的订阅

Apache Kafka 的流通过软件订阅提供。要管理您的订阅，请访问红帽客户门户中的帐户。

访问您的帐户

1. 转至 access.redhat.com。
2. 如果您还没有帐户，请创建一个帐户。
3. 登录到您的帐户。

激活订阅

1. 转至 access.redhat.com。
2. 导航到 **My Subscriptions**。
3. 导航到 **激活订阅** 并输入您的 16 位激活号。

下载 Zip 和 Tar 文件

要访问 zip 或 tar 文件，请使用客户门户网站查找下载的相关文件。如果您使用 RPM 软件包，则不需要这一步。

1. 打开浏览器并登录红帽客户门户网站 **产品下载页面**，网址为 access.redhat.com/downloads。
2. 在 **INTEGRATION AND AUTOMATION** 目录中找到 **Apache Kafka for Apache Kafka** 的流。
3. 选择 Apache Kafka 产品所需的流。此时会打开 **Software Downloads** 页面。
4. 单击组件的 **Download** 链接。

使用 DNF 安装软件包

要安装软件包以及所有软件包的依赖软件包，请使用：

```
dnf install <package_name>
```

要从本地目录中安装之前下载的软件包，请使用：

```
dnf install <path_to_download_package>
```

更新于 2024-06-25