Red Hat Streams for Apache Kafka 2.7

使用 Apache Kafka 控制台的流

AMQ Streams 控制台支持部署 AMQ Streams。

Last Updated: 2024-05-29

Red Hat Streams for Apache Kafka 2.7 使用 Apache Kafka 控制台的流

AMQ Streams 控制台支持部署 AMQ Streams。

法律通告

摘要

将控制台连接到由 AMQ Streams 管理的 Kafka 集群，并使用它来监控和管理集群。

目录

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。

技术预览

Apache Kafka 控制台的流是一个技术预览。

技术预览功能不被红帽产品服务级别协议(SLA)支持，且可能无法完成。因此，红帽不推荐在生产环境中实施任何技术预览功能。此技术预览功能为您提供对即将推出的产品创新的早期访问，允许您在开发过程中测试并提供反馈。如需有关支持范围的更多信息，请参阅 技术预览功能支持范围。

第 1 章 使用 APACHE KAFKA 控制台的 STREAMS 管理 KAFKA 集群

Apache Kafka 控制台的 Streams 提供了一个用户界面，用于协助管理 Kafka 集群，通过其用户界面提供用于监控、管理和优化每个集群的实时见解。

第 2 章 将 APACHE KAFKA 控制台的 STREAMS 连接到 KAFKA 集群

将 Apache Kafka 控制台的 Streams 部署到与 Apache Kafka 的 Streams 管理的 Kafka 集群相同的 OpenShift 集群。使用流为 Apache Kafka 控制台提供的安装文件。

对于每个 Kafka 集群，用于安装集群的 **Kafka** 资源的配置需要以下内容：

- 控制台连接集群所需的足够授权。
- 控制台可用于连接到集群的 **路由** 监听程序。
- 启用 Prometheus 并且能够从集群中提取指标。
- 指标配置（通过 **ConfigMap**）以适合 Prometheus 的格式导出指标。

Apache Kafka 控制台的 Streams 需要 Kafka 用户，配置为 **KafkaUser** 自定义资源，以便控制台以身份验证和授权的用户身份访问集群。

当您配置 **KafkaUser** 身份验证和授权机制时，请确保它们与对应的 **Kafka** 配置匹配。

- **KafkaUser.spec.authentication** 匹配 **Kafka.spec.kafka.listeners[*].authentication**
- **KafkaUser.spec.authorization** 匹配 **Kafka.spec.kafka.authorization**

注意

Prometheus 必须被安装并配置为从 Kubernetes 和 Kafka 集群提取指标，并在控制台中填充指标图形。

先决条件

- 安装需要具有 **cluster-admin** 角色的 OpenShift 用户，如 **system:admin**。
- OpenShift 4.12 到 4.15 集群。
- 由 Streams for Apache Kafka 管理的 Kafka 集群在 OpenShift 集群中运行的。
- Prometheus Operator，必须是与为 OpenShift 监控部署的独立 Operator。
- **oc** 命令行工具已安装并配置为连接到 OpenShift 集群。
- 在控制台中会话管理和身份验证的 secret 值。
  您可以使用 OpenSSL TLS 管理工具生成值，如下所示：

```
SESSION_SECRET=$(LC_CTYPE=C openssl rand -base64 32)
echo "Generated SESSION_SECRET: $SESSION_SECRET"

NEXTAUTH_SECRET=$(LC_CTYPE=C openssl rand -base64 32)
echo "Generated NEXTAUTH_SECRET: $NEXTAUTH_SECRET"
```

使用 **openssl help** 来获取所用的选项的命令行描述。

除了为 Apache Kafka 控制台安装流的文件外，Apache Kafka 控制台的 Streams 还提供预配置的文件，以便为 Apache Kafka Operator、Prometheus Operator 和 Kafka 集群安装 Streams。在此过程中，我们假定已安装了 operator。

安装文件提供了设置和试用控制台的最快速方法，但您可以使用自己的 Apache Kafka 和 Prometheus 部署流。

流程

1. 下载并提取 Apache Kafka 控制台安装工件的 Streams。
   控制台可从 Apache Kafka 软件下载页面 的 Streams 获得 。

   文件包含控制台、Kafka 集群和 Prometheus 所需的部署配置。

2. 通过应用 Prometheus 安装文件，使用控制台所需的配置创建 Prometheus 实例：

   a. 编辑 **console-prometheus-server.clusterrolebinding.yaml** 文件中的 **${NAMESPACE}**，以使用 Prometheus 实例要安装到的命名空间：

   ```
   sed -i 's/${NAMESPACE}/"my-project"/g' <resource_path>/console-prometheus-server.clusterrolebinding.yaml
   ```

   例如，在此流程中，我们将安装到 **my-project** 命名空间。配置将 Prometheus 的角色与其服务帐户绑定。

   b. 通过按照以下顺序应用安装文件来创建 Prometheus 实例：

   ```
   # Prometheus security resources
   oc apply -n my-project -f <resource_path>/prometheus/console-prometheus-server.clusterrole.yaml
   oc apply -n my-project -f <resource_path>/prometheus/console-prometheus-server.serviceaccount.yaml
   oc apply -n my-project -f <resource_path>/prometheus/console-prometheus-server.clusterrolebinding.yaml

   # Prometheus PodMonitor and Kubernetes scrape configurations
   oc apply -n my-project -f <resource_path>/prometheus/kafka-resources.podmonitor.yaml
   oc apply -n my-project -f <resource_path>/prometheus/kubernetes-scrape-configs.secret.yaml

   # Prometheus instance
   oc apply -n my-project -f <resource_path>/prometheus/console-prometheus.prometheus.yaml
   ```

   实例名为 **console-prometheus**，用于连接控制台的服务的 URL 为 **http://prometheus-operated.my-project.svc.cluster.local:9090**，并且 **my-project** 从命名空间名称中获取。

   注意

   没有为 **console-prometheus** 实例部署路由，因为不需要从 OpenShift 集群外部访问它。

3. 创建和部署 Kafka 集群。

a. 如果您使用在 KRaft 模式下运行的 Kafka 集群的控制台，请在 **console-kafka-metrics.configmap.yaml** 文件中更新集群的指标配置：

- 取消注释 KRaft 相关指标配置。
- 注释掉 ZooKeeper 相关指标。

此文件包含控制台所需的指标配置。

b. 通过添加 ACL 类型来为 Kafka 集群提供授权访问来编辑 **console-kafka-user1.kafkauser.yaml** 文件中的 **KafkaUser** 自定义资源。
Kafka 用户至少需要以下 ACL 规则：

- **描述** 集群资源的**DescribeConfigs** 权限
- **所有 主题 资源的 读取、 DescribeConfigs 权限**
- **所有 组资源 的读取、 描述 权限**

用户授权设置示例

```
apiVersion: kafka.strimzi.io/v1beta2
kind: KafkaUser
metadata:
  name: console-cluster-user1
  labels:
    strimzi.io/cluster: console-kafka
spec:
  authentication:
    type: scram-sha-512
  authorization:
    type: simple
    acls:
      - resource:
          type: cluster
          name: ""
          patternType: literal
        operations:
          - Describe
          - DescribeConfigs
      - resource:
          type: topic
          name: "*"
          patternType: literal
        operations:
          - Read
          - Describe
          - DescribeConfigs
      - resource:
```

```
    type: group
    name: "*"
    patternType: literal
  operations:
    - Read
    - Describe
```

c.

编辑 console-kafka.kafka.yaml 文件以替换占位符：

```
sed -i 's/type: ${LISTENER_TYPE}/type: route/g' console-kafka.kafka.yaml
sed -i 's/\${CLUSTER_DOMAIN}/"'<my_router_base_domain>'"/g' console-kafka.kafka.yaml
```

此文件包含用于创建 Kafka 集群的 Kafka 自定义资源配置。

这些命令执行以下操作：

- 将 type: ${LISTENER_TYPE} 替换为 type: route。
- 将 ${CLUSTER_DOMAIN} 替换为指定 bootstrap 和 per-broker 服务使用的路由监听程序主机所需的基域值。默认情况下，路由 监听程序主机由 OpenShift 自动分配。但是，您可以通过指定主机来覆盖分配的路由主机。

另外，您可以将 示例配置复制到您自己的 Kafka 部署 中。

d.

通过按以下顺序应用安装文件来创建 Kafka 集群：

```
# Metrics configuration
oc apply -n my-project -f console-kafka-metrics.configmap.yaml

# Create the cluster
oc apply -n my-project -f console-kafka.kafka.yaml

# Create a user for the cluster
oc apply -n my-project -f <resource_path>/console-kafka-user1.kafkauser.yaml
```

如果您使用自己的 Kafka 集群，请应用更新的 Kafka 资源配置而不是 console-

kafka.kafka.yaml。

安装文件会创建一个 Kafka 集群以及 Kafka 用户以及控制台所需的指标配置，用于连接到您要通过控制台监控的每个 Kafka 集群和指标配置。每个 Kafka 用户都需要一个唯一名称。

4. 检查部署的状态：

```
oc get pods -n <my_console_namespace>
```

输出显示 Operator 和集群就绪度

```
NAME                    READY   STATUS   RESTARTS
strimzi-cluster-operator    1/1     Running  0
console-kafka-kafka-0       1/1     Running  0
console-kafka-kafka-1       1/1     Running  0
console-kafka-kafka-2       1/1     Running  0
prometheus-operator-...     1/1     Running  0
console-prometheus          1/1     Running  0
```

在这里，console-kafka 是集群的名称。

标识创建的 pod 的 pod ID。

在默认部署中，您要安装三个 pod。

READY 显示就绪/预期的副本数。当 STATUS 显示为 Running 时，部署成功。

5. 安装 Apache Kafka 控制台的流。

a. 编辑 console-server.clusterrolebinding.yaml 文件，以使用控制台实例要安装到的命名空间：

```
sed -i 's/${NAMESPACE}/"'"my-project"'"/g' /<resource_path>console-
server.clusterrolebinding.yaml
```

配置将控制台的角色与其服务帐户绑定。

b. 通过应用安装文件（按以下顺序）安装控制台用户界面并路由到接口：

```
# Console security resources
oc apply -n my-project -f <resource_path>/console-server.clusterrole.yaml
oc apply -n my-project -f <resource_path>/console-server.serviceaccount.yaml
oc apply -n my-project -f <resource_path>/console-server.clusterrolebinding.yaml

# Console user interface service
oc apply -n my-project -f <resource_path>/console-ui.service.yaml

# Console route
oc apply -n my-project -f <resource_path>/console-ui.route.yaml
```

安装会创建运行控制台用户界面所需的角色、角色绑定、服务帐户、服务和路由。

c. 为控制台中的会话管理和身份验证，创建一个名为 console-ui-secrets 的 Secret，其中包含两个 secret 值（如先决条件中所述）：

```
oc create secret generic console-ui-secrets -n my-project \
  --from-literal=SESSION_SECRET="<session_secret_value>" \
  --from-literal=NEXTAUTH_SECRET="<next_secret_value>"
```

部署控制台时，secret 作为环境变量挂载。

d. 获取为控制台用户界面创建的路由的主机名：

```
oc get route console-ui-route -n my-project -o jsonpath='{.spec.host}'
```

访问控制台用户界面需要主机名。

e. 编辑 console.deployment.yaml 文件以替换占位符：

```
sed -i 's/${CONSOLE_HOSTNAME}/"<route_hostname>"/g'
console.deployment.yaml
sed -i 's/${NAMESPACE}/"my-project"/g' console.deployment.yaml
```

这些命令执行以下操作：

- 将 https://${CONSOLE_HOSTNAME} 替换为 https://<route_hostname >，这是用于访问控制台用户界面的路由。
- 将 ${NAMESPACE} 替换为 http://prometheus-operated.${NAMESPACE}.svc.cluster.local:9090 中的 my-project 命名空间名称，这是控制台使用的 Prometheus 实例的 URL。

如果您使用自己的 Kafka 集群，请确保使用正确的 值配置 环境变量。这些值可让控制台连接到集群并检索指标。

f. 安装控制台 API：

```
oc apply -n my-project -f <resource_path>/console.deployment.yaml
```

输出显示控制台就绪

```
NAME                    READY   STATUS   RESTARTS
strimzi-cluster-operator    1/1     Running  0
console-kafka-kafka-0       1/1     Running  0
console-kafka-kafka-0       1/1     Running  0
console-kafka-kafka-0       1/1     Running  0
prometheus-operator-...     1/1     Running  0
console-prometheus          1/1     Running  0
console-api                 1/1     Running  0
console-ui                  1/1     Running  0
```

在您自己的 Kafka 集群中添加示例配置

如果您已经安装了 Kafka 集群，您可以使用所需的配置更新 Kafka 资源。在应用集群配置文件时，请使用更新的 Kafka 资源，而不是使用由 Apache Kafka Console 安装文件提供的 Kafka 资源。

Kafka 资源需要以下配置：

- 用于公开集群以进行控制台连接 的路由 监听程序
- 为检索集群中的指标启用 Prometheus 指标。如果您使用 ZooKeeper 进行元数据管理，请为 ZooKeeper 添加相同的配置。

Prometheus 指标配置必须引用 ConfigMap，该 ConfigMap 提供控制台所需的指标配置。指标配置在 console-cluster-metrics.configmap.yaml 资源配置文件中提供。

控制台连接的 Kafka 集群配置示例

```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: console-kafka
  namespace: my-project
spec:
  entityOperator:
    topicOperator: {}
    userOperator: {}
  kafka:
    authorization:
      type: simple
    config:
      allow.everyone.if.no.acl.found: 'true'
      default.replication.factor: 3
      inter.broker.protocol.version: '3.6'
      min.insync.replicas: 2
      offsets.topic.replication.factor: 3
      transaction.state.log.min.isr: 2
      transaction.state.log.replication.factor: 3
    listeners: 1
      - name: route1
        port: 9094
        tls: true
        type: route
        authentication:
          type: scram-sha-512
    replicas: 3
    storage:
      type: jbod
      volumes:
      - id: 0
        type: persistent-claim
```

```
      size: 10Gi
      deleteClaim: false
  metricsConfig: (2)
    type: jmxPrometheusExporter
    valueFrom:
      configMapKeyRef:
        name: console-cluster-metrics
        key: kafka-metrics-config.yml
  version: 3.6.0
zookeeper:
  replicas: 3
  storage:
    deleteClaim: false
    size: 10Gi
    type: persistent-claim
  metricsConfig: (3)
    type: jmxPrometheusExporter
    valueFrom:
      configMapKeyRef:
        name: console-cluster-metrics
        key: zookeeper-metrics-config.yml
```

(1)

为控制台连接公开集群的监听程序。在本例中，配置了路由监听程序。

(2)

Prometheus 指标，通过引用包含 Prometheus JMX 导出器配置的 ConfigMap 启用。

(3)

只有在使用带有 ZooKeeper 的 Apache Kafka 的 Streams for Apache Kafka 时才添加 ZooKeeper 配置。KRaft 模式不需要它。

检查控制台部署环境变量

如果使用自己的 Kafka 集群，请检查控制台的部署配置是否有所需的环境变量。

以下前缀决定了环境变量值的范围：

- KAFKA 代表所有 Kafka 集群的配置。

- CONSOLE_KAFKA_<UNIQUE_NAME_ID_FOR_CLUSTER > 代表每个特定集群的配置。

控制台部署配置示例

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: console
spec:
  replicas: 1
  # ...
  template:
    metadata:
      labels:
        app: console
    spec:
      # ...
      containers:
      - name: console-api
        # ...
        env:
        - name: KAFKA_SECURITY_PROTOCOL (1)
          value: SASL_SSL
        - name: KAFKA_SASL_MECHANISM (2)
          value: SCRAM-SHA-512
        - name: CONSOLE_KAFKA_CLUSTER1 (3)
          value: my-project/console-kafka
        - name: CONSOLE_KAFKA_CLUSTER1_BOOTSTRAP_SERVERS (4)
          value: console-kafka-route1-bootstrap-my-project.router.com:443
        - name: CONSOLE_KAFKA_CLUSTER1_SASL_JAAS_CONFIG (5)
          valueFrom:
            secretKeyRef:
              name: console-cluster-user1
              key: sasl.jaas.config
      - name: console-ui
        # ...
        env:
        - name: NEXTAUTH_SECRET (6)
          valueFrom:
            secretKeyRef:
              name: console-ui-secrets
              key: NEXTAUTH_SECRET
        - name: SESSION_SECRET (7)
          valueFrom:
            secretKeyRef:
              name: console-ui-secrets
              key: SESSION_SECRET
        - name: NEXTAUTH_URL (8)
          value: 'https://console-ui-route-my-project.router.com'
        - name: BACKEND_URL (9)
```

```
    value: 'http://127.0.0.1:8080'
  - name: CONSOLE_METRICS_PROMETHEUS_URL 10
    value: 'http://prometheus-operated.my-project.svc.cluster.local:9090'
```

1

用于与 Kafka 代理通信的安全协议。

2

Kafka 代理的控制台（客户端）身份验证的 SASL 机制。

3

在 Kafka 资源配置中为集群指定的命名空间和名称。

4

bootstrap 代理地址的主机和端口对，用于发现并连接到 Kafka 集群中的所有代理。在本例中，使用路由监听程序地址。侦听器在 Kafka 资源中配置。

5

作为 Secret 挂载到 Apache Kafka 控制台部署的 Kafka 用户的身份验证凭据。secret 中的 sasl.jaas.config 属性包含 SASL 凭据，如用户名和密码。此处，凭据作为 Secret 挂载。

6

用于在控制台中进行身份验证的 secret。

7

在控制台中会话管理的 secret

8

连接到 Apache Kafka 用户界面和用户访问控制台的流的 URL。

9

控制台用户界面与 进行通信的后端服务器，以进行数据检索。

10

第 3 章 APACHE KAFKA 控制台的 STREAMS

当您打开 Apache Kafka 控制台的 Streams 时，主页会显示连接的 Kafka 集群列表。点击主页上的 Kafka 集群名称或从侧边菜单中点 Kafka 集群名称，您可以找到有关以下组件的信息：

Kafka 集群

一组 Kafka 代理和管理组件。

代理（Broker）

代理包含主题，编配存储并传递信息。

topics

主题提供数据存储的目的地。Kafka 将每个主题分成一个或多个分区。

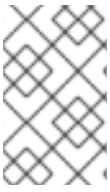
分区

用于数据分片和复制的主题子集。分区数量在主题配置中定义。

消费者组

Kafka 组具有相同组 ID 的用户，并在组成员间分发信息。组中的消费者从一个或多个分区接收数据。

例如，您可以在进入到查看集群代理和主题的信息或连接到 Kafka 集群的消费者组前查看 Kafka 集群的状态。



注意

如果侧边菜单不可见，请单击控制台标头中的 hamburger 菜单（三个水平行）。

第 4 章 HOME ：检查连接的集群

主页提供连接的 Kafka 集群的快照，提供代理状态和关联消费者组计数。在探索主题时，主页中可以方便地显示有关您最近的主题视图的详细信息。

要查找更多信息：

- 点集群名称在 Cluster overview 页面中查找集群指标。
- 单击最近查看的主题，以检索有关该特定主题的详细信息。

第 5 章 集群概览页面

Cluster overview 页面显示 Kafka 集群的状态。在这里，您可以评估 Kafka 代理的就绪情况，识别任何集群错误或警告，并深入了解集群的健康状况。页面概览中提供了有关集群中主题和分区数量的信息，以及它们的复制状态。通过图表探索集群指标，显示已用磁盘空间、CPU 使用率和内存使用情况。另外，主题指标提供了对 Kafka 集群中所有主题的总传入和传出字节率的全面视图。

5.1. 访问客户端访问的集群连接详情

将客户端连接到 Kafka 集群时，按照以下步骤从集群概览页面检索必要的连接详情。

流程

1. 在 Apache Kafka 控制台的 Streams 中，点您要连接的 Kafka 集群的名称，然后点 Cluster overview 和 Cluster connection details。

2. 在 Kafka 客户端配置中复制并添加 bootstrap 地址和连接属性，以与 Kafka 集群建立连接。

注意

确保客户端使用的身份验证类型与为 Kafka 集群配置的身份验证类型匹配。

第 6 章 主题页面

主题 页面显示为 Kafka 集群创建的所有主题。使用此页面检查有关主题的信息。

主题 页面显示主题中分区的整体复制状态，以及主题中分区的数量以及关联的消费者组的数量。该主题使用的整体存储也显示。

警告

内部主题不能修改。您可以从主题页面中返回的主题列表中选择隐藏内部 主题。

点击主题名称，在一系列标签页中会显示其他主题信息：

消息

消息显示主题的消息日志。

分区

分区显示主题中每个分区的复制状态。

消费者组

消费者组列出了连接到主题的消费者组和组成员的名称和状态。

配置

配置显示主题的配置。

如果一个主题显示为 Managed，这意味着使用 Apache Kafka Topic Operator 的 Streams 管理，且不会直接在 Kafka 集群中创建。

使用选项卡提供的信息，检查和修改主题的配置。

6.1. 检查主题消息

从 Messages 选项卡中跟踪特定主题的消息流。Messages 选项卡显示主题的按时间顺序列表。

流程

1. 在 Apache Kafka 控制台的 Streams 中，点 Kafka 集群的名称，然后点 Topics。
2. 点您要检查的主题名称。
3. 检查 消息 选项卡上的信息。

   对于每个消息，您可以看到其时间戳（以 UTC）、偏移、键和值。

   单击一条消息，您可以看到完整的消息详情。

   单击 Manage 列 图标（代表两列），以选择要显示的信息。
4. 单击搜索下拉菜单，再选择高级搜索选项来优化您的搜索。

   选择显示来自指定时间或偏移的最新消息。您可以显示所有分区或指定分区的信息。

   完成后，您可以点击 CSV 图标（代表 CSV 文件）下载有关返回的信息。

拒绝搜索

在这个示例中，搜索术语和消息、检索和分区选项被合并：

- messages=timestamp:2024-03-01T00:00:00Z retrieve=50 partition=1 Error on page load where=value

过滤器在分区 1 中搜索文本"Error on page load"作为消息值，从 2024 年 3 月 1 日开始，并检索到 50 个消息。

搜索术语

输入搜索词作为文本(*包含单词*)以查找特定匹配项，并在消息中定义查找术语 *的位置*。您可以在消息中的任何位置搜索，或者将搜索范围缩小到键、标头或值。

例如：

- messages=latest retrieve=100 642-26-1594 where=key

本例在消息键 642-26-1594 中搜索最新的 100 消息。

消息选项

设置返回消息的起点。

- latest 从最新的消息开始。
  - messages=latest
- 以 ISO 8601 格式从准确时间和日期开始的时间戳。
  - messages=timestamp:2024-03-14T00:00:00Z
- 从分区中的偏移开始的偏移量。在某些情况下，您可能想要指定没有分区的偏移。但是，最常见的场景是通过特定分区中的偏移搜索。
  - messages=offset:5600253 partition=0
- UNIX Timestamp 从 Unix 格式的时间和日期开始。
  - messages=epoch:1

检索选项

设置检索选项。

- 返回指定数量的消息数。
  - messages=latest retrieve=50
- 持续 实时返回最新消息。点 pause 按钮（由两个垂直线代表）暂停刷新。取消暂停以继续刷新。
  - retrieve=continuously

分区选项

选择针对所有分区或特定分区运行搜索。

6.2. 检查主题分区

从 Partitions 选项卡中检查特定主题的分区。Partitions 选项卡显示属于某个主题的分区列表。

流程

1. 在 Apache Kafka 控制台的 Streams 中，点 Kafka 集群的名称，然后点 Topics。
2. 点您要从 主题 页面检查的主题名称。
3. 检查 分区 选项卡上的信息。

对于每个分区，您可以看到其复制状态，以及指定分区领导、副本代理以及分区存储的数据量的信息。

您可以通过复制状态查看分区：

in-sync

主题中的所有分区都是完全复制的。当副本(followers)与指定的分区领导处于"in-sync"时，分区是完全复制的。如果副本已获取了在允许滞后时间内领导分区的日志结束偏移，则副本为 'in-sync'，由 replica.lag.time.max.ms 决定。

under-replicated

如果某些副本(followers)没有同步，分区会被复制。复制不足的状态信号数据复制中潜在的问题。

Offline

主题中的一些或者所有分区当前都不可用。这可能是因为需要调查和解决的代理失败或网络问题等问题。

您还可以检查指定为分区领导和包含副本的代理的信息：

leader

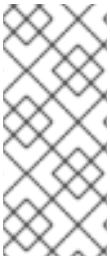
领导机处理所有生成请求。在其他代理上遵循其他代理复制领导的数据。如果与领导的最新提交消息捕获，则后续者被视为 in-sync。

首选领导

在创建新主题时，Kafka 的领导选举算法从每个分区的副本列表中分配一个领导机。该算法旨在平衡领导分配分布。"是"值表示当前领导机是首选领导机，建议平衡领导力分布。"No"值可能在领导分配中没有平衡，需要进一步调查。如果分区的领导分配没有良好平衡，它可以贡献大小差异。平衡良好的 Kafka 集群应该在代理间均匀分布领导角色。

Replicas

复制领导的数据的遵循者。副本提供容错和数据可用性。



注意

在代理间分布数据的差异可能代表 Kafka 集群中的平衡问题。如果某些代理一致处理大量数据，这可能表示分区不会在代理中均匀分布。这可能会导致资源利用率不均匀，并可能会影响这些代理的性能。

6.3. 检查主题消费者组

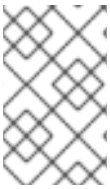
从 Consumer groups 选项卡中检查特定主题的消费者组。Consumer groups 选项卡显示与主题关联的消费者组列表。

流程

1. 在 Apache Kafka 控制台的 Streams 中，点 Kafka 集群的名称，然后点 Topics。
2. 点您要从 主题 页面检查的主题名称。
3. 检查 Consumer groups 选项卡的信息。
4. 若要检查消费者组成员，请单击使用者组名称。

对于每个消费者组，您可以看到其状态、所有分区的整体消费者滞后以及成员数。有关检查消费者组的详情，请参考 第 8 章 *消费者组页面*。

对于每个组成员，您会看到分配给消费者组、整个消费者滞后以及分配的分区数量的唯一(consumer) 客户端 ID。有关检查消费者组成员的详情，请参考 第 8.1 节 “检查消费者组成员”。



注意

监控消费者组行为对于确保消费者之间的消息优化分布非常重要。

6.4. 检查主题配置

从 Configuration 选项卡中检查特定主题的配置。Configuration 选项卡显示该主题的配置值列表。

流程

1. 在 Apache Kafka 控制台的 Streams 中，点 Kafka 集群的名称，然后点 Topics。
2. 点您要从 主题 页面检查的主题名称。
3. 检查 Configuration 选项卡上的信息。

您可以过滤要检查的属性，包括通过数据源选择：

- DEFAULT_CONFIG 属性具有预定义的默认值。当没有用于这些属性的用户定义的值时，会使用这个值。
- STATIC_BROKER_CONFIG 属性具有适用于整个代理且扩展到该代理管理的所有主题的预定义值。当没有用于这些属性的用户定义的值时，会使用这个值。
- DYNAMIC_TOPIC_CONFIG 属性值已被为特定主题配置，并覆盖默认的配置值。

提示

Apache Kafka Topic Operator 的 Streams 简化了使用 KafkaTopic 资源管理 Kafka 主题的过程。

第 7 章 代理页面

Brokers 页面显示为 Kafka 集群创建的所有代理。对于每个代理，您可以看到其状态，以及代理间分区分布，包括分区领导和副本的数量。

代理状态显示为以下之一：

稳定

稳定代理正常运行，且没有严重问题。

unstable

不稳定的代理可能会遇到问题，如大量资源使用量或网络问题。

如果代理有一个机架 ID，这是代理所在的机架或数据中心的 ID。

点击代理名称旁边的右箭头(>)查看代理的更多信息，包括其主机名和磁盘用量。

注意

如果分布不均匀，请考虑重新平衡，以确保资源利用率效率。

第 8 章 消费者组页面

Consumer groups 页面显示与 Kafka 集群关联的所有消费者组。对于每个消费者组，您可以看到其状态、所有分区的整体消费者滞后以及成员数。单击相关的主题，以显示主题 页面选项卡中可用的主题 信息。

消费者组状态可以是以下之一：

- stable 表示正常正常工作
- 重新平衡 表示消费者组成员的持续调整。
- 空 建议没有活跃的成员。如果处于空状态，请考虑向组添加成员。

点消费者组名称来检查组成员。有关检查消费者组成员的详情，请参考 第 8.1 节 “检查消费者组成员”。

8.1. 检查消费者组成员

从 Consumer groups 页面检查特定消费者组的成员。

流程

1. 在 Apache Kafka 控制台的 Streams 中，点 Kafka 集群的名称，然后点 Consumer groups。
2. 点您要从 Consumer groups 页面检查的消费者组的名称。
3. 点成员 ID 旁边的右箭头(>)来查看与成员关联的主题分区，以及任何可能的消费者滞后。

对于每个组成员，您会看到分配给消费者组、整个消费者滞后以及分配的分区数量的唯一(consumer)客户端 ID。

特定主题分区的任何消费者滞后反映消费者获取的最后一条消息之间的差距（提交偏移位置）和由制作者（偏移位置）编写的最新消息。

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 2024-04-30