



Red Hat AMQ Streams 2.5

在 AMQ Streams Kafka Bridge 中使用 3scale API 管理

使用 3scale 提供的功能和功能

Red Hat AMQ Streams 2.5 在 AMQ Streams Kafka Bridge 中使用 3scale API 管理

使用 3scale 提供的功能和功能

法律通告

Copyright © 2023 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

在 OpenShift Container Platform 中部署 AMQ Streams Kafka Bridge，并集成 Red Hat 3scale API Management。

目录

使开源包含更多	3
第 1 章 3SCALE API 管理	4
1.1. KAFKA BRIDGE 服务发现	4
1.2. 3SCALE APICAST 网关策略	4
1.3. 3SCALE APICAST 用于 TLS 验证	6
1.4. 使用现有的 3SCALE 部署	6
第 2 章 为 KAFKA BRIDGE 部署 3SCALE	7
附录 A. 使用您的订阅	11
访问您的帐户	11
激活订阅	11
下载 Zip 和 Tar 文件	11
使用 DNF 安装软件包	11

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅[我们的首席技术官 Chris Wright 提供的消息](#)。

第 1 章 3SCALE API 管理

如果在 OpenShift Container Platform 上部署了 Kafka Bridge，您可以在 3scale 中使用它。

对于 Kafka Bridge 的普通部署，没有置备身份验证或授权，且不支持外部客户端的 TLS 加密连接。3scale API 管理可以使用 TLS 保护 Kafka Bridge，并提供身份验证和授权。与 3scale 集成还意味着可以使用 metrics、速率限制和计费额外功能。

使用 3scale 时，您可以将不同类型的身份验证用于来自希望访问 AMQ Streams 的外部客户端。3scale 支持以下验证类型：

标准 API 密钥

单一随机字符串或哈希充当标识符和机密令牌。

应用程序标识符和密钥对

不可变标识符和可变 secret key 字符串。

OpenID Connect

委派的身份验证的协议。

1.1. KAFKA BRIDGE 服务发现

3scale 使用服务发现功能集成，这需要 3scale 部署到与 AMQ Streams 和 Kafka Bridge 相同的 OpenShift 集群。

您的 AMQ Streams Cluster Operator 部署必须设置以下环境变量：

- STRIMZI_CUSTOM_KAFKA_BRIDGE_SERVICE_LABELS
- STRIMZI_CUSTOM_KAFKA_BRIDGE_SERVICE_ANNOTATIONS

当部署了 Kafka Bridge 时，用于公开 Kafka Bridge 的 REST 接口的服务使用注解和标签以供 3scale 发现。

- 3scale 使用 **discovery.3scale.net=true** 标签来查找服务。
- 注解提供有关该服务的信息。

您可以在 OpenShift 控制台中通过导航到 Kafka Bridge 实例的 **Services** 来检查您的配置。在 **Annotations** 下，您会看到 Kafka Bridge 的 OpenAPI 规格的端点。

1.2. 3SCALE APICAST 网关策略

3scale 与 3scale APIcast 结合使用，3scale APIcast 是一个与 3scale 一起部署的 API 网关，为 Kafka Bridge 提供单一入口点。

APIcast 策略提供了一种机制来定制网关的运行方式。3scale 为网关配置提供一组标准策略。您还可以创建自己的策略。

有关 APIcast 策略的更多信息，请参阅 [Red Hat 3scale 文档](#)。

Kafka Bridge 的 APIcast 策略

3scale 与 Kafka Bridge 集成的示例策略配置随 **policies_config.json** 文件提供，该文件定义：

- 匿名访问

- 标头修改
- 路由
- URL rewriting

网关策略通过此文件启用或禁用。

您可以使用此示例来定义您自己的策略。

匿名访问

匿名访问策略在没有身份验证的情况下公开服务，在 HTTP 客户端不提供它们时提供默认凭证（匿名访问）。该策略不是强制性的，如果始终需要身份验证，则可以禁用或删除。

标头修改

标头修改策略允许修改现有的 HTTP 标头，或向通过网关的请求或响应添加新的标头。对于 3scale 集成，策略会向从 HTTP 客户端通过网关传递给 Kafka 网桥的每个请求添加标头。

当 Kafka Bridge 收到创建一个新消费者的请求时，它会返回一个 JSON 有效负载，其中包含一个带有消费者所有后续请求必须使用的 URI 的 **base_uri**。例如：

```
{
  "instance_id": "consumer-1",
  "base_uri": "http://my-bridge:8080/consumers/my-group/instances/consumer1"
}
```

使用 APIcast 时，客户端会将所有后续请求发送到网关，而不是直接发送到 Kafka Bridge。因此，URI 需要网关主机名，而不是网关后面的 Kafka 网桥的地址。

使用标头修改策略，将标头添加到来自 HTTP 客户端的请求，以便 Kafka Bridge 使用网关主机名。

例如，通过应用 **Forwarded: host=my-gateway:80;proto=http** 标头，Kafka Bridge 会为使用者提供以下内容。

```
{
  "instance_id": "consumer-1",
  "base_uri": "http://my-gateway:80/consumers/my-group/instances/consumer1"
}
```

X-Forwarded-Path 标头将来自客户端的请求中包含的原始路径传输到网关。当网关支持多个 Kafka Bridge 实例时，这个标头严格与应用的路由策略相关。

路由

当存在多个 Kafka Bridge 实例时，会应用路由策略。请求必须发送到初始创建使用者的同一 Kafka Bridge 实例，因此请求必须为网关指定一个路由，以将请求转发到适当的 Kafka Bridge 实例。

路由策略会命名每个网桥实例，并使用名称执行路由。在部署 **Kafka Bridge** 时，您可以在 KafkaBridge 自定义资源中指定名称。

例如，每个请求（使用 **X-Forwarded-Path**）从消费者到：

http://my-gateway:80/my-bridge-1/consumers/my-group/instances/consumer1

转发到：

http://my-bridge-1-bridge-service:8080/consumers/my-group/instances/consumer1

URL 重写策略会删除网桥名称，因为它在将请求从网关转发到 Kafka Bridge 时不使用它。

URL rewriting

在将请求从网关转发到 Kafka Bridge 时，URL rewiring 策略确保从客户端到特定 Kafka Bridge 的请求不包含网桥名称。

网桥名称不在网桥公开的端点中使用。

1.3. 3SCALE APICAST 用于 TLS 验证

您可以为 TLS 验证设置 APICast，它需要使用模板自管理 APICast。 **apicast** 服务作为路由公开。

您还可以在 Kafka Bridge API 中应用 TLS 策略。

1.4. 使用现有的 3SCALE 部署

如果您已将 3scale 部署到 OpenShift，并且希望将其与 Kafka Bridge 搭配使用，请确保您以完成了[K为 Kafka Bridge 部署 3scale](#)中介绍的设置。

第 2 章 为 KAFKA BRIDGE 部署 3SCALE

要将 3scale 与 Kafka Bridge 搭配使用，您首先部署它，然后将其配置为发现 Kafka Bridge API。

您还要使用 3scale APIcast 和 3scale toolbox。

- APIcast 由 3scale 提供，作为一个基于 NGINX 的 API 网关提供，供 HTTP 客户端连接到 Kafka Bridge API 服务。
- 3scale toolbox 是一个配置工具，用于将 Kafka Bridge 服务的 OpenAPI 规格导入到 3scale。

在这种情况下，您将在同一 OpenShift 集群中运行 AMQ Streams、Kafka、Kafka Bridge 和 3scale/APIcast。



注意

如果您已在与 Kafka Bridge 相同的集群中部署了 3scale，您可以跳过部署步骤并使用当前部署。

先决条件

- 了解 3scale
- [AMQ Streams 和 Kafka 正在运行](#)
- [部署了 Kafka Bridge](#)

对于 3scale 部署：

- 检查 [Red Hat 3scale API Management 支持的配置](#)。
- 安装需要具有 **cluster-admin** 角色的用户，如 **system:admin**。
- 您需要访问描述以下内容的 JSON 文件：
 - Kafka Bridge OpenAPI specification (**openapiv2.json**)
 - Kafka Bridge 的标头修改和路由策略(**policies_config.json**)
在 [GitHub](#) 上查找 JSON 文件。

如需更多信息，请参阅 [Red Hat 3scale 文档](#)。

流程

1. 将 3scale API Management 部署到 OpenShift 集群。
 - a. 创建新项目或使用现有项目。

```
oc new-project my-project \
  --description="description" --display-name="display_name"
```

- b. 部署 3scale。
[Red Hat 3scale 文档](#) 介绍了如何使用模板或 operator 在 OpenShift 上部署 3scale。

无论您使用哪种方法，请确保将 `WILDCARD_DOMAIN` 参数设置为 OpenShift 集群的域。

记录访问 3scale 管理门户的 URLs 和凭证。

- 为 3scale 授予 3scale 授权来发现 Kafka Bridge 服务：

```
oc adm policy add-cluster-role-to-user view system:serviceaccount:my-project:amp
```

- 通过 OpenShift 控制台或 CLI 验证 3scale 已成功部署到 OpenShift 集群。
例如：

```
oc get deployment 3scale-operator
```

- 设置 3scale toolbox。

- 根据 [Red Hat 3scale 文档](#) 中的信息来安装 3scale toolbox。
- 设置环境变量以便与 3scale 交互：

```
export REMOTE_NAME=stirzi-kafka-bridge ①
export SYSTEM_NAME=stirzi_http_bridge_for_apache_kafka ②
export TENANT=stirzi-kafka-bridge-admin ③
export PORTAL_ENDPOINT=$TENANT.3scale.net ④
export TOKEN=3scale access token ⑤
```

- REMOTE_NAME** 是分配给 3scale 管理门户的远程地址的名称。
- SYSTEM_NAME** 是通过 3scale toolbox 导入 OpenAPI 规格创建的 3scale 服务/API 的名称。
- TENANT** 是 3scale 管理门户的租户名称（即 [https://\\$TENANT.3scale](https://$TENANT.3scale)）。
- PORTAL_ENDPOINT** 是运行 3scale 管理门户的端点。
- TOKEN** 是 3scale 管理门户提供的访问令牌，用于通过 3scale toolbox 或 HTTP 请求进行交互。

- 配置 3scale toolbox 的远程 web 地址：

```
3scale remote add $REMOTE_NAME https://$TOKEN@$PORTAL_ENDPOINT/
```

现在，每次运行 toolbox 时不需要指定 3scale 管理门户的端点地址。

- 检查您的 Cluster Operator 部署是否有 3scale 发现 Kafka Bridge 服务所需的标签和注解属性。

```
#...
env:
- name: STRIMZI_CUSTOM_KAFKA_BRIDGE_SERVICE_LABELS
  value: |
    discovery.3scale.net=true
- name: STRIMZI_CUSTOM_KAFKA_BRIDGE_SERVICE_ANNOTATIONS
  value: |
    discovery.3scale.net/scheme=http
    discovery.3scale.net/port=8080
```

```
discovery.3scale.net/path=/
discovery.3scale.net/description-path=/openapi
#...
```

如果没有，请通过 OpenShift 控制台添加属性，或尝试重新部署 [Cluster Operator](#) 和 [Kafka Bridge](#)。

6. 通过 3scale 发现 Kafka Bridge API 服务。

- a. 使用部署 3scale 时提供的凭证，登录 3scale 管理门户。
- b. 在管理门户仪表板的 APIs 中，点 **Create Product**。
- c. 点 **Import from OpenShift**。
- d. 选择 Kafka Bridge 服务
- e. 点 **Create Product**。
您可能需要刷新页面来查看 Kafka Bridge 服务。

现在，您需要导入该服务的配置。您从编辑器中执行此操作，但保持门户已打开，以检查导入是否成功。

7. 编辑 OpenAPI 规格(JSON 文件)中的 **Host** 字段，以使用 Kafka Bridge 服务的基本 URL：
例如：

```
"host": "my-bridge-bridge-service.my-project.svc.cluster.local:8080"
```

检查 **host** URL 包括正确的：

- Kafka Bridge 名称 (*my-bridge*)
- 项目名称 (*my-project*)
- Kafka Bridge 的端口 (*8080*)

8. 使用 3scale toolbox 导入更新的 OpenAPI 规格：

```
3scale import openapi -k -d $REMOTE_NAME openapiv2.json -t myproject-my-bridge-bridge-service
```

9. 为服务导入标头修改和路由策略 (JSON 文件)。

- a. 找到您在 3scale 中创建的服务的 ID。
在这里，我们使用 ['jq' 实用程序](#)：

```
export SERVICE_ID=$(curl -k -s -X GET
"https://$PORTAL_ENDPOINT/admin/api/services.json?access_token=$TOKEN" | jq
".services[] | select(.service.system_name | contains(\"$SYSTEM_NAME\")) |
.service.id")
```

导入策略时需要 ID。

- b. 导入策略：

```
curl -k -X PUT
"https://$PORTAL_ENDPOINT/admin/api/services/$SERVICE_ID/proxy/policies.json" --
data "access_token=$TOKEN" --data-urlencode policies_config@policies_config.json
```

10. 从 3scale 管理门户，导航到 **Integration → Configuration**，以检查 Kafka Bridge 服务的端点和策略是否已加载。
11. 导航到 **Applications → Create Application Plan** 以创建应用程序计划。
12. 进入到 **Audience → Developer → Applications → Create Application** 来创建一个应用程序。需要应用程序才能获取用于身份验证的用户密钥。
13. （生产环境步骤）要在生产环境的网关中使用 API，请提升配置：

```
3scale proxy-config promote $REMOTE_NAME $SERVICE_ID
```

14. 使用 API 测试工具来验证您可以通过 APICast 网关访问 Kafka 网桥，方法是使用调用来创建消费者，以及为应用程序创建的用户密钥。
例如：

```
https://my-project-my-bridge-bridge-service-3scale-apicast-
staging.example.com:443/consumers/my-group?
user_key=3dfc188650101010ecd7fdc56098ce95
```

如果从 Kafka Bridge 返回有效负载，则已成功创建使用者。

```
{
  "instance_id": "consumer1",
  "base uri": "https://my-project-my-bridge-bridge-service-3scale-apicast-
staging.example.com:443/consumers/my-group/instances/consumer1"
}
```

Base URI 是客户端在后续请求中使用的地址。

附录 A. 使用您的订阅

AMQ Streams 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 **AMQ Streams for Apache Kafka** 项。
3. 选择所需的 AMQ Streams 产品。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 2023-11-22