



Red Hat Fuse 7.13

在 OpenShift 上管理 Fuse

使用 Fuse 控制台管理 Fuse 应用程序

Red Hat Fuse 7.13 在 OpenShift 上管理 Fuse

使用 Fuse 控制台管理 Fuse 应用程序

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

当您部署 Fuse 应用程序时，您可以使用 Fuse 控制台来监控并与红帽 Fuse 集成进行交互。

目录

前言	3
使开源包含更多	4
第 1 章 关于 FUSE 控制台	5
第 2 章 在 OPENSIFT 4.X 中设置 FUSE 控制台	6
2.1. 使用 OPERATORHUB 在 OPENSIFT 4.X 上安装和部署 FUSE 控制台	6
2.2. 使用命令行在 OPENSIFT 4.X 上安装和部署 FUSE 控制台	9
2.3. OPENSIFT 4.X 上 FUSE 控制台的基于角色的访问控制	14
2.4. 在 OPENSIFT 4.X 上升级 FUSE 控制台	19
2.5. 在 OPENSIFT 4.X 服务器上升级 FUSE 镜像流和模板	21
2.6. 在 OPENSIFT 4.X 上调整 FUSE 控制台的性能	24
第 3 章 在 OPENSIFT 3.11 中设置 FUSE 控制台	31
3.1. 在 OPENSIFT 3.11 上部署 FUSE 控制台	32
3.2. 从 OPENSIFT 3.11 上的 FUSE 控制台监控单个 FUSE POD	34
第 4 章 查看容器和应用程序	36
第 5 章 查看并管理 APACHE CAMEL 应用程序	37
5.1. 启动、暂停或删除上下文	37
5.2. 查看 CAMEL 应用程序详情	38
5.3. 查看 CAMEL 路由列表并与之交互	39
5.4. 调试路由	41
第 6 章 查看 AMQ BROKER	43
第 7 章 查看并管理 JMX 域和 MBEANS	44
第 8 章 查看并管理 QUARTZ SCHEDULES	46
第 9 章 查看诊断	47
第 10 章 查看线程	48
第 11 章 确保数据在 FUSE 控制台中正确显示	49
附录 A. FUSE 控制台配置属性	50

前言

Red Hat Fuse 提供两个企业监控工具，用于查看和管理 Fuse 集成：

- Fuse 控制台是一个基于 Web 的控制台，您可以从浏览器访问以监控和管理正在运行的 Fuse 容器。Fuse 控制台基于 Hawtio 开源软件(<https://hawt.io/>)。本指南论述了如何使用 Fuse 控制台。
- Prometheus 为 Fuse 分发存储系统和集成级别指标。您可以使用 Grafana 等图形分析接口来查看和分析存储的历史数据。如需了解更多有关使用 Prometheus 的信息，请参阅以下文档：
 - [Prometheus 文档](#)
 - [OpenShift 上的 Fuse 指南](#)
 - [在 OpenShift Container Platform 上安装和操作 Fuse Online](#)

本指南的受众是 JBoss EAP 管理员上的红帽 Fuse。本指南假定您熟悉红帽 Fuse 平台、Apache Camel 以及您所在机构的处理要求。

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。这些更改将在即将发行的几个发行本中逐渐实施。详情请查看我们的 [CTO Chris Wright 信息](#)。

第 1 章 关于 FUSE 控制台

红帽 Fuse 控制台是一个基于 Hawtio 开源软件的 Web 控制台。如需支持的浏览器列表，请转至 [支持的配置](#)。

Fuse 控制台提供了一个中央接口，用于检查和管理一个或多个部署的 Fuse 容器的详细信息。您还可以监控红帽 Fuse 和系统资源，执行更新，以及启动或停止服务。

当您在 OpenShift 上安装 Red Hat Fuse standalone 或使用 Fuse 时，可以使用 Fuse 控制台。您可以在 Fuse 控制台中查看和管理的集成取决于正在运行的插件。可能的插件包括：

- Camel
- JMX
- OSGI
- Runtime
- 日志

第 2 章 在 OPENSIFT 4.X 中设置 FUSE 控制台

在 OpenShift 4.x 上，设置 Fuse 控制台涉及安装和部署它。您有安装和部署 Fuse 控制台的这些选项：

- [第 2.1 节 “使用 OperatorHub 在 OpenShift 4.x 上安装和部署 Fuse 控制台”](#)
您可以使用 Fuse Console Operator 安装和部署 Fuse 控制台，以便它能够访问特定命名空间中的 Fuse 应用程序。Operator 会为您处理 Fuse 控制台的安全。
- [第 2.2 节 “使用命令行在 OpenShift 4.x 上安装和部署 Fuse 控制台”](#)
您可以使用命令行和其中一个 Fuse 控制台模板来安装和部署 Fuse 控制台，以便它可以访问 OpenShift 集群或特定命名空间中的多个命名空间中的 Fuse 应用程序。在部署前，您必须通过生成客户端证书来保护 Fuse 控制台。

另外，您可以为 Fuse 控制台自定义基于角色的访问控制(RBAC)，如 [第 2.3 节 “OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制”](#) 所述。

2.1. 使用 OPERATORHUB 在 OPENSIFT 4.X 上安装和部署 FUSE 控制台

要在 OpenShift 4.x 上安装 Fuse 控制台，您可以使用 OpenShift OperatorHub 提供的 Fuse Console Operator。要部署 Fuse 控制台，请创建一个已安装的 Operator 实例。

先决条件

- 您已使用 [registry.redhat.io](#) 配置身份验证，如 [为容器镜像使用 registry.redhat.io 验证](#) 中所述。
- 如果要为 Fuse 控制台自定义基于角色的访问控制(RBAC)，则必须在安装 Fuse Console Operator 的同一 OpenShift 命名空间中有一个 RBAC 配置映射文件。如果要使用默认的 RBAC 行为，如 [OpenShift 4.x 上 Fuse Console 的基于角色的访问控制](#) 中所述，您不需要提供配置映射文件。

流程

安装和部署 Fuse 控制台：

1. 以具有集群管理员访问权限的用户身份登录 Web 浏览器中的 OpenShift 控制台。
2. **点 Operators，然后点 OperatorHub。**
3. **在搜索字段窗口中，键入 Fuse Console 来过滤操作器列表。**
4. **点 Fuse Console Operator。**
5. **在 Fuse Console Operator 安装窗口中，点 Install。**

Create Operator Subscription 表单将打开。

- 对于 **更新频道**，请选择 **7.13.x**。
- 对于 **Installation Mode**，接受默认值（集群中的特定命名空间）。

请注意，在安装 Operator 后，当部署 Fuse 控制台时，您可以选择监控集群中的所有命名空间中的应用程序，或者仅在安装了 Fuse Console operator 的命名空间中监控应用程序。

- 对于 **Installed Namespace**，选择要在其中安装 Fuse Console Operator 的命名空间。
- 对于 **Update Approval**，您可以选择 **Automatic** 或 **Manual** 来配置 OpenShift 如何处理 Fuse Console Operator 的更新。
 - 如果选择 **Automatic** 更新，当有新版本的 Fuse Console Operator 可用时，OpenShift Operator Lifecycle Manager (OLM) 将自动升级 Fuse 控制台的运行实例，而无需人为干预。
 - 如果选择 **手动** 更新，则当有新版 Operator 可用时，OLM 会创建更新请求。作为集群管理员，您必须手动批准该更新请求，才可将 Fuse Console Operator 更新至新版本。

6. 点 **Install**。

OpenShift 在当前命名空间中安装 Fuse Console Operator。

7. 要验证安装，点 **Operators**，然后点 **Installed Operators**。您可以在操作器列表中看到 Fuse 控制台。

8. 使用 OpenShift Web 控制台部署 Fuse 控制台：

- a. 在 **Installed Operators** 列表中，单击 **Name** 列下的 **Fuse Console**。

- b. 在 **Provided APIs** 下的 **Operator Details** 页面中，点 **Create Instance**。

接受配置默认值或选择性地编辑它们。

对于 **Replicas**，如果要提高 Fuse 控制台性能（例如在高可用性环境中），您可以增加分配给 Fuse 控制台的 pod 数量。

对于 **Rbac**（基于角色的访问控制），只有在 **config Map** 字段中指定一个值，如果要自定义默认的 RBAC 行为，并且是否安装了 **Fuse Console Operator** 的命名空间中已存在 **ConfigMap** 文件。如需有关 RBAC 的更多信息，请参阅 [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)。


对于 **Nginx**，请参阅 [Fuse Console Operator 安装的性能调优](#)。

- c. 点 **Create**。

Fuse Console Operator Details 页面将打开并显示部署状态。

9. 打开 Fuse 控制台：

- a. 对于 **命名空间 部署**：在 **OpenShift Web 控制台** 中，打开安装 **Fuse Console operator** 的项目，然后选择 **Overview**。在 **Project Overview** 页面中，向下滚动到 **Launcher** 部分，再单击 **Fuse Console** 链接。

对于 **集群部署**，在 **OpenShift Web 控制台** 的标题栏中，点网格图标()。在弹出菜单中，单击 **红帽应用程序** 下的 **Fuse 控制台 URL** 链接。

- b. 登录到 **Fuse 控制台**。

浏览器中打开了 **Authorize Access** 页面，其中列出了所需权限。

- c. 点击 **Allow selected permissions**。

Fuse 控制台在浏览器中打开，并显示您有权访问的 Fuse 应用程序 pod。

10. 对于您要查看的应用程序，点 **Connect**。

此时将打开一个新浏览器窗口，显示 Fuse 控制台中的应用程序。

2.2. 使用命令行在 OPENSIFT 4.X 上安装和部署 FUSE 控制台

在 OpenShift 4.x 中，您可以选择其中一个部署选项来从命令行安装和部署 Fuse 控制台：

- **集群** - Fuse 控制台可以发现并连接到在 OpenShift 集群上的多个命名空间（项目）间部署的 Fuse 应用程序。要部署此模板，您必须具有 OpenShift 集群的管理员角色。
- **具有基于角色的访问控制的集群** - 具有可配置的基于角色的访问控制(RBAC)的集群模板。如需更多信息，请参阅 [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)。
- **命名空间** - Fuse 控制台可以访问特定的 OpenShift 项目（命名空间）。要部署此模板，您必须具有 OpenShift 项目的管理员角色。
- **带有基于角色的访问控制的命名空间** - 带有可配置 RBAC 的命名空间模板。如需更多信息，请参阅 [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)。

要查看 Fuse 控制台模板的参数列表，请运行以下命令：

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-namespace-os4.json
```

先决条件

- 在安装和部署 Fuse 控制台前，您必须生成一个使用服务签名证书颁发机构签名的客户端证书，如生成证书以保护 [OpenShift 4.x 上的 Fuse 控制台安全](#) 中所述。
- 有 OpenShift 集群的集群管理员角色。

- 您已使用 `registry.redhat.io` 配置身份验证，如 [为容器镜像使用 registry.redhat.io 验证](#) 中所述。
- 已安装 Fuse Console 镜像流（以及其他 Fuse 镜像流），如在 [OpenShift 4.x 服务器上安装 Fuse 镜像流和模板](#) 中所述。

流程

1. 使用以下命令验证是否安装了 Fuse Console 镜像流，以检索所有模板的列表：

```
oc get template -n openshift
```

2. 另外，如果要使用新发行标签更新已安装的镜像流，请使用以下命令将 Fuse Console 镜像导入到 `openshift` 命名空间：

```
oc import-image fuse7/fuse-console-rhel8:1.10 --from=registry.redhat.io/fuse7/fuse-console-rhel8:1.10 --confirm -n openshift
```

3. 运行以下命令来获取 Fuse Console APP_NAME 值：

```
oc process --parameters -f TEMPLATE-FILENAME
```

其中 `TEMPLATE-FILENAME` 是以下模板之一：

- 集群模板：

https://github.com/jboss-fuse/application-templates/blob/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-cluster-os4.json
- 使用可配置的 RBAC 的集群模板：

https://github.com/jboss-fuse/application-templates/blob/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-cluster-rbac.yml
- 命名空间模板：

https://github.com/jboss-fuse/application-templates/blob/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-namespaces-os4.json

- 带有可配置的 RBAC 的命名空间模板：

https://github.com/jboss-fuse/application-templates/blob/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-namespaces-rbac.yml

例如，对于使用可配置的 RBAC 的集群模板，请运行以下命令：

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-cluster-rbac.yml
```

4.

从您在 [OpenShift 4.x 上保护 Fuse 控制台](#) 生成的证书，使用以下命令创建 `secret` 并将其挂载到 Fuse 控制台中（其中 `APP_NAME` 是 Fuse Console 应用程序的名称）。

```
oc create secret tls APP_NAME-tls-proxying --cert server.crt --key server.key
```

5.

运行以下命令，基于 Fuse Console 模板的本地副本创建新应用程序（其中 `myproject` 是 OpenShift 项目的名称，`mytemp` 是包含 Fuse Console 模板的本地目录的路径，`myhost` 是用于访问 Fuse 控制台的主机名）：

- 对于集群模板：

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-cluster-os4.json -p ROUTE_HOSTNAME=myhost
```

- 对于使用 RBAC 模板的集群：

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-cluster-rbac.yml -p ROUTE_HOSTNAME=myhost
```

- 对于命名空间模板：

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-namespace-os4.json
```

- 对于使用 **RBAC** 模板的命名空间：

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fuse-console-namespace-rbac.yml
```

6. 要配置 Fuse 控制台使其能够打开 OpenShift Web 控制台，请运行以下命令来设置 **OPENSIFT_WEB_CONSOLE_URL** 环境变量：

```
oc set env dc/${APP_NAME} OPENSIFT_WEB_CONSOLE_URL=`oc get -n openshift-config-managed cm console-public -o jsonpath={.data.consoleURL}`
```

7. 运行以下命令，获取 Fuse 控制台部署的状态和 URL：

```
oc status
```

8. 要从浏览器访问 Fuse 控制台，请使用在第 7 步中返回的 URL（例如 <https://fuse-console.192.168.64.12.nip.io>）。

2.2.1. 生成证书来保护 OpenShift 4.x 上的 Fuse 控制台

在 OpenShift 4.x 上，若要在 Fuse 控制台代理和 Jolokia 代理之间保持连接，必须在部署 Fuse 控制台前生成客户端证书。服务签名证书颁发机构私钥必须用于为客户端证书签名。

只有在您要使用命令行安装和部署 Fuse 控制台时，才必须按照以下步骤操作。如果使用 Fuse Console Operator，它会为您处理此任务。



重要

您必须为每个 OpenShift 集群生成并签署单独的客户端证书。对于多个集群，不要使用相同的证书。

先决条件

- 有 **集群管理员** 对 **OpenShift 集群** 的访问权限。
- 如果您要为多个 **OpenShift 集群** 生成证书，并且之前在当前目录中为不同的集群生成证书，请执行以下操作之一以确保为当前集群生成不同的证书：
 - 从当前目录中删除现有证书文件（如 **ca.crt**、**ca.key** 和 **ca.srl**）。
 - 更改到其他工作目录。例如，如果您的当前工作目录命名为 **cluster1**，请创建一个新的 **cluster2** 目录，并将工作目录改为其中：

```
mkdir ../cluster2
```

```
cd ../cluster2
```

流程

1. 以具有**集群管理员**访问权限的用户身份登录 **OpenShift**：

```
oc login -u <user_with_cluster_admin_role>
```

2. 执行以下命令来检索**服务签名证书颁发机构**密钥：

- 检索证书：

```
oc get secrets/signing-key -n openshift-service-ca -o "jsonpath={.data['tls.crt']}" | base64 --decode > ca.crt
```

- 检索私钥：

```
oc get secrets/signing-key -n openshift-service-ca -o "jsonpath={.data['tls.key']}" | base64 --decode > ca.key
```

3. 生成客户端证书，如 [Kubernetes 证书管理](#) 中所述，使用 **easysrsa**、**openssl** 或 **cfssl**。

以下是使用 `openssl` 的示例命令：

- a. 生成私钥：

```
openssl genrsa -out server.key 2048
```

- b. 编写 CSR 配置文件。

```
cat <<EOT >> csr.conf
[ req ]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn

[ dn ]
CN = fuse-console.fuse.svc

[ v3_ext ]
authorityKeyIdentifier=keyid,issuer:always
keyUsage=keyEncipherment,dataEncipherment,digitalSignature
extendedKeyUsage=serverAuth,clientAuth
EOT
```

在这里，`CN` 参数中的值指的是应用程序使用的命名空间。

- c. 生成 CSR：

```
openssl req -new -key server.key -out server.csr -config csr.conf
```

- d. 发布签名证书：

```
openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt
-days 10000 -extensions v3_ext -extfile csr.conf
```

后续步骤

您需要此证书来为 **Fuse 控制台** 创建 `secret`，如 [使用命令行在 OpenShift 4.x 上安装和部署 Fuse 控制台](#) 中所述。

2.3. OPENSIFT 4.X 上 FUSE 控制台的基于角色的访问控制

Fuse 控制台提供基于角色的访问控制(RBAC)，它根据 OpenShift 提供的用户授权推断访问。在 Fuse 控制台中，RBAC 决定用户在 pod 上执行 MBean 操作。

如需有关 OpenShift 授权的信息，请参阅 [OpenShift 文档中的使用 RBAC 定义和应用权限](#) 部分。

当使用 Operator 在 OpenShift 上安装 Fuse 控制台时，默认启用基于角色的访问控制。

如果要使用模板安装 Fuse 控制台，为它实施基于角色的访问权限，则必须使用可通过 RBAC 配置的模板之一(`fuse-console-cluster-rbac.yml` 或 `fuse-console-namespace-rbac.yml`)来安装 Fuse 控制台，如 [使用命令行在 OpenShift 4.x 上安装和部署 Fuse 控制台](#)。

Fuse 控制台 RBAC 利用 OpenShift 中容器集资源的操作动词访问，以确定用户在 Fuse 控制台中对 pod 的 MBean 操作的访问。默认情况下，Fuse 控制台有两个用户角色：

- **admin**

如果用户可以在 OpenShift 中更新 pod，则用户会限制 Fuse 控制台的 admin 角色。用户可以在 Fuse 控制台中为 pod 执行写入 MBean 操作。

- **viewer**

如果用户可以在 OpenShift 中获取 pod，则用户会限制 Fuse 控制台的 viewer 角色。用户可以在 Fuse 控制台中为 pod 执行只读 MBean 操作。



注意

如果您使用非RBAC 模板安装 Fuse 控制台，则只有授予 pod 资源更新动词的 OpenShift 用户有权执行 Fuse Console MBeans 操作。授予 pod 资源的 get 动词的用户可以查看 pod，但不能执行任何 Fuse 控制台操作。

其他资源

- [确定 OpenShift 4.x 上 Fuse 控制台的访问角色](#)

- [在 OpenShift 4.x 上自定义对 Fuse 控制台进行基于角色的访问权限](#)
- [在 OpenShift 4.x 上为 Fuse 控制台禁用基于角色的访问控制](#)

2.3.1. 确定 OpenShift 4.x 上 Fuse 控制台的访问角色

Fuse 控制台基于角色的访问控制是从用户的 pod 的 OpenShift 权限中推断出来的。要确定向特定用户授予的 Fuse 控制台访问角色，请获取授予 Pod 用户的 OpenShift 权限。

先决条件

- 您知道用户名。
- 您知道 pod 的名称。

流程

- 要确定用户是否有用于 pod 的 Fuse Console admin 角色，请运行以下命令来查看用户是否可以更新 OpenShift 上的 pod：

```
oc auth can-i update pods/<pod> --as <user>
```

如果响应是 **yes**，则用户具有 pod 的 Fuse 控制台 管理员角色。用户可以在 Fuse 控制台中为 pod 执行写入 MBean 操作。

- 要确定用户是否有用于 pod 的 Fuse Console viewer 角色，请运行以下命令来查看用户是否可以在 OpenShift 上获取 pod：

```
oc auth can-i get pods/<pod> --as <user>
```

如果响应是 **yes**，则用户具有 pod 的 Fuse Console viewer 角色。用户可以在 Fuse 控制台中为 pod 执行只读 MBean 操作。根据上下文，Fuse 控制台可防止具有 viewer 角色的用户执行写入 MBean 操作，方法是禁用一个选项，或者在用户尝试写入 MBean 操作时显示"operation not allowed for this user"消息。

如果没有，则用户不绑定到任何 Fuse 控制台角色，用户无法在 Fuse 控制台中查看 pod。

其他资源

- [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)
- [在 OpenShift 4.x 上自定义对 Fuse 控制台进行基于角色的访问权限](#)
- [在 OpenShift 4.x 上为 Fuse 控制台禁用基于角色的访问控制](#)

2.3.2. 在 OpenShift 4.x 上自定义对 Fuse 控制台进行基于角色的访问权限

如果您使用 OperatorHub 安装 Fuse 控制台，则默认启用 [基于角色的访问控制\(RBAC\)](#)，如 [OpenShift 4.x 上 Fuse Console](#) 的基于角色的访问控制所述。如果要自定义 Fuse Console RBAC 行为，在部署 Fuse 控制台前，您必须提供 ConfigMap 文件（定义自定义 RBAC 行为）。您必须将自定义 ConfigMap 文件放在安装 Fuse Console Operator 的同一命名空间中。

如果您使用命令行模板安装 Fuse 控制台，则 deployment-cluster-rbac.yml 和 deployment-namespace-rbac.yml 模板会创建一个包含配置文件(ACL.yml)的 ConfigMap。配置文件定义 MBean 操作允许的角色。

前提条件

- 您可以使用 OperatorHub 或其中一个 Fuse 控制台 RBAC 模板(deployment-cluster-rbac.yml 或 deployment-namespace-rbac.yml)安装 Fuse 控制台。

流程

自定义 Fuse 控制台 RBAC 角色：

1. 如果使用命令行安装 Fuse 控制台，安装模板会包含默认的 ConfigMap 文件，以便您可以跳过下一步。

如果您在部署 Fuse 控制台前，使用 OperatorHub 安装 Fuse 控制台，则创建一个 RBAC ConfigMap：

- a.

确保当前的 OpenShift 项目是您要安装 Fuse 控制台的项目。例如，如果您要在 `fusetest` 项目中安装 Fuse 控制台，请运行以下命令：

```
oc project fusetest
```

b.

要从模板创建 **Fuse Console RBAC ConfigMap** 文件，请运行以下命令：

```
oc process -f {sb2-templates-base-url}/fuse-console-operator-rbac.yml -p  
APP_NAME=fuse-console | oc create -f -
```

2.

运行以下命令，在编辑器中打开 **ConfigMap**：

```
oc edit cm $APP_NAME-rbac
```

例如：

```
oc edit cm fuse-console-rbac
```

3.

编辑该文件。

4.

保存文件以使改变生效。OpenShift 会自动重启 **Fuse 控制台 Pod**。

其他资源

- [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)
- [确定 OpenShift 4.x 上 Fuse 控制台的访问角色](#)
- [在 OpenShift 4.x 上为 Fuse 控制台禁用基于角色的访问控制](#)

2.3.3. 在 OpenShift 4.x 上为 Fuse 控制台禁用基于角色的访问控制

如果您使用命令行安装 Fuse 控制台，并且指定了 Fuse 控制台 RBAC 模板之一，**Fuse Console** 的 `HAWTIO_ONLINE_RBAC_ACL` 环境变量会将基于角色的访问控制(RBAC) **ConfigMap** 配置文件路径传递给 **OpenShift** 服务器。如果没有指定 `HAWTIO_ONLINE_RBAC_ACL` 环境变量，则禁用 RBAC 支

持，并且只有被授予 pod 资源(OpenShift 中)更新 动词的用户才会在 Fuse 控制台中调用 pod 的 MBeans 操作。

请注意，当您使用 OperatorHub 安装 Fuse 控制台时，默认启用基于角色的访问控制，并且不会应用 HAWTIO_ONLINE_RBAC_ACL 环境变量。

前提条件

您可以使用命令行安装 Fuse 控制台，并指定其中一个 Fuse 控制台 RBAC 模板(deployment-cluster-rbac.yml 或 deployment-namespace-rbac.yml)。

流程

为 Fuse 控制台禁用基于角色的访问控制：

1. 在 OpenShift 中，编辑 Fuse 控制台的 Deployment Config 资源。
2. 删除整个 HAWTIO_ONLINE_RBAC_ACL 环境变量定义。

(请注意，仅清除其值并不够)。
3. 保存文件以使改变生效。OpenShift 会自动重启 Fuse 控制台 Pod。

其他资源

- [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)
- [确定 OpenShift 4.x 上 Fuse 控制台的访问角色](#)
- [在 OpenShift 4.x 上自定义对 Fuse 控制台进行基于角色的访问权限](#)

2.4. 在 OPENSIFT 4.X 上升级 FUSE 控制台

Red Hat OpenShift 4.x 处理对 Operator 的更新，包括 Red Hat Fuse operator。如需更多信息，请参阅 [Operator OpenShift 文档](#)。

另外，Operator 更新可以根据应用程序的配置方式触发应用程序升级。

对于 Fuse 控制台应用程序，您还可以通过编辑应用程序自定义资源定义的 `.spec.version` 字段来触发对应用程序的升级。

前提条件

- 有 OpenShift 集群管理员权限。

流程

升级 Fuse 控制台应用程序：

1. 在终端窗口中，使用以下命令更改应用程序自定义资源定义的 `.spec.version` 字段：

```
oc patch -n <project-name> <custom-resource-name> --type='merge' -p '{"spec": {"version":"1.7.1"}}'
```

例如：

```
oc patch -n myproject hawtio/example-fuseconsole --type='merge' -p '{"spec": {"version":"1.7.1"}}'
```

2. 检查应用程序的状态是否已更新：

```
oc get -n myproject hawtio/example-fuseconsole
```

响应显示有关应用程序的信息，包括版本号：

```
NAME          AGE  URL
example-fuseconsole 1m  https://fuseconsole.192.168.64.38.nip.io
docker.io/fuseconsole/online:1.7.1
```

当您更改 `.spec.version` 字段的值时，OpenShift 会自动重新部署应用。

3.

检查版本更改触发的重新部署状态：

```
oc rollout status deployment.v1.apps/example-fuseconsole
```

成功部署会显示这个响应：

```
deployment "example-fuseconsole" successfully rolled out
```

2.5. 在 OPENSIFT 4.X 服务器升级 FUSE 镜像流和模板

OpenShift Container Platform 4.x 使用 Samples Operator，它在 OpenShift 命名空间中运行，升级和更新基于 Red Hat Enterprise Linux (RHEL) 的 OpenShift Container Platform 镜像流和模板。

升级 OpenShift 镜像流和模板上的 Fuse：

- 重新配置 Samples Operator
- 将 Fuse 镜像流和模板添加到 Skipped Imagestreams 和 Skipped Templates 字段中。
 - 跳过的 Imagestreams：位于 Samples Operator 清单中的 Imagestreams，但集群管理员希望 Operator 忽略或不管理。
 - 跳过的模板：位于 Samples Operator 清单中的模板，但集群管理员希望 Operator 忽略或不予管理。

先决条件

- 您可以访问 OpenShift 服务器。
- 您已将身份验证配置为 registry.redhat.io。

流程

1. 启动 **OpenShift 4 服务器**。

2. 以管理员身份登录 **OpenShift 服务器**。

```
oc login --user system:admin --token=my-token --server=https://my-cluster.example.com:6443
```

3. 验证您是否使用您为其创建 **docker-registry secret** 的项目。

```
oc project openshift
```

4. 查看 **Samples operator** 的当前配置。

```
oc get configs.samples.operator.openshift.io -n openshift-cluster-samples-operator -o yaml
```

5. 配置 **Samples operator** 以忽略添加的 **fuse 模板和镜像流**。

```
oc edit configs.samples.operator.openshift.io -n openshift-cluster-samples-operator
```

6. 添加 **Fuse imagestreams Skipped Imagestreams** 部分，并将 **Fuse 和 Spring Boot 2 模板** 添加到 **Skipped Templates** 部分。

```
[...]
spec:
  architectures:
  - x86_64
  managementState: Managed
  skippedImagestreams:
  - fuse-console-rhel8
  - fuse-eap-openshift-jdk8-rhel7
  - fuse-eap-openshift-jdk11-rhel8
  - fuse-java-openshift-rhel8
  - fuse-java-openshift-jdk11-rhel8
  - fuse-karaf-openshift-rhel8
  - fuse-karaf-openshift-jdk11-rhel8
  - fuse-apicurito-generator-rhel8
  - fuse-apicurito-rhel8
  skippedTemplates:
  - s2i-fuse713-eap-camel-amq
  - s2i-fuse713-eap-camel-cdi
  - s2i-fuse713-eap-camel-cxf-jaxrs
  - s2i-fuse713-eap-camel-cxf-jaxws
  - s2i-fuse713-karaf-camel-amq
```

```

- s2i-fuse713-karaf-camel-log
- s2i-fuse713-karaf-camel-rest-sql
- s2i-fuse713-karaf-cxf-rest
- s2i-fuse713-spring-boot-2-camel-amq
- s2i-fuse713-spring-boot-2-camel-config
- s2i-fuse713-spring-boot-2-camel-drools
- s2i-fuse713-spring-boot-2-camel-infinispan
- s2i-fuse713-spring-boot-2-camel-rest-3scale
- s2i-fuse713-spring-boot-2-camel-rest-sql
- s2i-fuse713-spring-boot-2-camel
- s2i-fuse713-spring-boot-2-camel-xa
- s2i-fuse713-spring-boot-2-camel-xml
- s2i-fuse713-spring-boot-2-cxf-jaxrs
- s2i-fuse713-spring-boot-2-cxf-jaxws
- s2i-fuse713-spring-boot-2-cxf-jaxrs-xml
- s2i-fuse713-spring-boot-2-cxf-jaxws-xml

```

7.

在 OpenShift 镜像流上升级 Fuse。

```

BASEURL=https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001

```

```

oc replace -n openshift -f ${BASEURL}/fis-image-streams.json

```

8.

在 OpenShift 快速启动模板上升级 Fuse :

```

for template in eap-camel-amq-template.json \
eap-camel-cdi-template.json \
eap-camel-cxf-jaxrs-template.json \
eap-camel-cxf-jaxws-template.json \
karaf-camel-amq-template.json \
karaf-camel-log-template.json \
karaf-camel-rest-sql-template.json \
karaf-cxf-rest-template.json ;
do
oc replace -n openshift \
${BASEURL}/quickstarts/${template}
done

```

9.

升级 Spring Boot 2 快速启动模板 :

```

for template in spring-boot-2-camel-amq-template.json \
spring-boot-2-camel-config-template.json \
spring-boot-2-camel-drools-template.json \
spring-boot-2-camel-infinispan-template.json \
spring-boot-2-camel-rest-3scale-template.json \
spring-boot-2-camel-rest-sql-template.json \
spring-boot-2-camel-template.json \
spring-boot-2-camel-xa-template.json \

```

```
spring-boot-2-camel-xml-template.json \  
spring-boot-2-cxf-jaxrs-template.json \  
spring-boot-2-cxf-jaxws-template.json \  
spring-boot-2-cxf-jaxrs-xml-template.json \  
spring-boot-2-cxf-jaxws-xml-template.json ;  
do oc replace -n openshift \  
${BASEURL}/quickstarts/${template}  
done
```

10.

(可选) 查看 OpenShift 模板上的升级 Fuse :

```
oc get template -n openshift
```

2.6. 在 OPENSIFT 4.X 上调整 FUSE 控制台的性能

默认情况下，Fuse 控制台使用以下 Nginx 设置：

- **clientBodyBufferSize: 256k**
- **proxyBuffers: 16 128k**
- **subrequestOutputBufferSize: 10m**

注：有关这些设置的描述，请参阅 Nginx 文档：<http://nginx.org/en/docs/dirindex.html>

要调整 Fuse 控制台的性能，您可以设置任何 `clientBodyBufferSize`、`proxyBuffers` 和 `subrequestOutputBufferSize` 环境变量。例如，如果您使用 Fuse 控制台监控大量 pod 和路由（例如，共 100 个路由），您可以通过将 Fuse Console 的 `subrequestOutputBufferSize` 环境变量设置为在 60m 到 100m 之间来解决加载超时问题。

如何设置这些环境变量取决于您在 OpenShift 4.x 上安装 Fuse 控制台：

- 使用 Fuse Console Operator
- 通过使用 Fuse Console 模板

2.6.1. Fuse Console Operator 安装的性能调整

在 Openshift 4.x 上，您可以在部署 Fuse 控制台之前或之后设置 Nginx 性能调优环境变量。如果您随后这样做，OpenShift 会重新部署 Fuse 控制台。

先决条件

- 有集群管理员对 OpenShift 集群的访问权限。
- 您已安装 Fuse Console Operator，如使用 [OperatorHub](#) 在 OpenShift 4.x 上安装和部署 Fuse 控制台 中所述。

流程

您可以在部署 Fuse 控制台之前或之后设置环境变量。

- 在部署 Fuse 控制台前设置环境变量：
 1. 在 OpenShift Web 控制台中，安装有 Fuse Console Operator 的项目中，选择 Operators > Installed Operators > Red Hat Integration - Fuse Console。
 2. 单击 **Hawtio** 选项卡，然后单击 **Create Hawtio**。
 3. 在 **Create Hawtio** 页面中，在 **Form** 视图中，向下滚动到 **Config > Nginx** 部分。
 4. 展开 **Nginx** 部分，然后设置环境变量。例如：
 - **clientBodyBufferSize: 256k**
 - **proxyBuffers: 16 128k**
 - **subrequestOutputBufferSize: 100m**

5. 保存配置。
6. 点 **Create** 以部署 Fuse 控制台。
7. 部署完成后，打开 **Deployments> fuse-console** 页面，然后点 **Environment** 来验证环境变量是否在列表中。

- 在部署 Fuse 控制台后设置环境变量：

1. 在 OpenShift Web 控制台中，打开部署 Fuse 控制台的项目。
2. 选择 **Operators> Installed Operators> Red Hat Integration - Fuse Console**。
3. 单击 **Hawtio** 选项卡，然后单击 **fuse-console**。
4. 选择 **Actions> Edit Hawtio**。
5. 在 **Editor** 窗口中，滚动到 **spec** 部分。
6. 在 **spec** 部分，添加新的 **nginx** 部分并指定一个或多个环境变量，例如：

```
apiVersion: hawt.io/v1alpha1
kind: Hawtio
metadata:
  name: fuse-console
spec:
  type: Namespace
  nginx:
    clientBodyBufferSize: 256k
    proxyBuffers: 16 128k
    subrequestOutputBufferSize: 100m
  .
  .
  .
```

7. 点击 **Save**。

OpenShift 重新部署 Fuse 控制台。
8. 重新部署完成后，打开 **Workloads > Deployments > fuse-console** 页面，然后点击 **Environment** 来查看列表中的环境变量。

2.6.2. Fuse Console 模板安装的性能调整

在 Openshift 4.x 上，您可以在部署 Fuse 控制台之前或之后设置 Nginx 性能调优环境变量。如果您随后这样做，OpenShift 会重新部署 Fuse 控制台。

先决条件

- 有 **集群管理员** 对 **OpenShift 集群** 的访问权限。
- 您已在 **OpenShift** 上安装 **Fuse 控制台模板**，如在 [OpenShift 4.x 服务器上安装 Fuse 镜像流和模板](#) 中所述。

流程

您可以在部署 **Fuse 控制台** 之前或之后设置环境变量。

- 在部署 **Fuse 控制台** 前设置环境变量：
 1. 确定您要使用的 **Fuse 控制台模板**：
 - **集群模板(fuse-console-cluster-os4.json)**
 - **带有可配置的 RBAC 的集群模板(fuse-console-cluster-rbac.yml)**
 - **命名空间模板(fuse-console-namespace-os4.json)**

- 带有可配置的 RBAC (fuse-console-namespace-rbac.yml)的命名空间模板
2. 编辑您要用于 Fuse 控制台的 Fuse Console 模板的本地副本，使其包含 `NGINX_CLIENT_BODY_BUFFER_SIZE`、`NGINX_PROXY_BUFFERS` 和/或 `NGINX_SUBREQUEST_OUTPUT_BUFFER_SIZE` 环境变量，如下例所示：

```

apiVersion: apps.openshift.io/v1
kind: DeploymentConfig
metadata:
  name: fuse-console
spec:
  template:
    spec:
      containers:
      - env:
        - name: NGINX_CLIENT_BODY_BUFFER_SIZE
          value: 256k
        - name: NGINX_PROXY_BUFFERS
          value: 16 128k
        - name: NGINX_SUBREQUEST_OUTPUT_BUFFER_SIZE
          value: 100m

```

3. 保存您的更改。
4. 按照 [在 OpenShift 4.x 中设置 Fuse 控制台](#) 中所述，请按照安装和部署 Fuse 控制台的步骤进行操作。

- 在部署 Fuse 控制台后设置环境变量：

1. 在终端窗口中，登录到 OpenShift 集群。
2. 打开部署 Fuse 控制台的项目。例如，如果在 `myfuse` 项目中部署了 Fuse 控制台，请使用以下命令：

```

oc project myfuse

```
3. 获取 Fuse 控制台部署的名称：

oc get deployments

此命令返回在当前项目中运行的部署列表。例如：

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
fuse-console	1/1	1	1	114m

4.

运行以下命令，为 **Fuse** 控制台部署设置环境变量：

```
oc set env dc/fuse-console NGINX_CLIENT_BODY_BUFFER_SIZE="256k"
oc set env dc/fuse-console NGINX_PROXY_BUFFERS="16 128k"
oc set env dc/fuse-console NGINX_SUBREQUEST_OUTPUT_BUFFER_SIZE="10m"
```

OpenShift 重新部署 Fuse 控制台。

5.

重新部署完成后，验证环境变量设置：

a.

获取 **Fuse Console** pod 名称：

```
oc get pods
```

b.

运行以下命令来查看环境设置

```
oc exec <fuse-console-podname> -- cat /opt/app-root/etc/nginx.d/nginx-gateway.conf | grep "Performance tuning" -A 3
```

例如，如果 pod 名称是 **fuse-console-6646cbbd4c-9rplg**，请运行以下命令：

```
oc exec fuse-console-6646cbbd4c-9rplg -- cat /opt/app-root/etc/nginx.d/nginx-gateway.conf | grep "Performance tuning" -A 3
```

2.6.3. 在 Fuse 控制台中查看应用程序的性能调整

通过增强的 **Fuse** 控制台性能调优功能，您可以查看具有大量 **MBeans** 的应用程序。要使用此功能，请执行以下步骤。

先决条件

- 有 **集群管理员** 对 **OpenShift 集群** 的访问权限。
- 您已安装 **Fuse Console Operator**，如使用 [OperatorHub](#) 在 **OpenShift 4.x** 上安装和部署 **Fuse 控制台** 中所述。

流程

1. 增加应用程序的内存限值。

需要增加内存限值，例如从 256Mi 增加到 512 Mi，因此应用程序在到达 **Fuse 控制台** 前不会造成 OOM 错误崩溃。对于 **Fuse 快速入门**，请编辑应用程序的 `src/main/jkube/deployment.yml` 文件。

```
spec:
  template:
    spec:
      containers:
      -
        resources:
          [...]
          limits:
            cpu: "1.0"
            memory: 512Mi
```

2. 确保 **Fuse Console Deployment** 或 **DeploymentConfig** 具有足够的内存限值。如果还不够，将限制从 200Mi 增加到 512Mi。
3. 如果您在 `nginx` 日志中看到 `"too big subrequest 响应 while to client"` 错误，请应用 [第 2.6.1 节 “Fuse Console Operator 安装的性能调整”](#) 部分中提到的解决方案。

第 3 章 在 OPENSIFT 3.11 中设置 FUSE 控制台

在 OpenShift 3.11 中，您可以访问 Fuse 控制台：

- 通过将 Fuse 控制台添加到 OpenShift 项目，以便您可以监控项目中所有正在运行的 Fuse 容器。
- 通过将 Fuse 控制台添加到 OpenShift 集群，以便您可以监控集群中的所有项目中正在运行的 Fuse 容器。
- 从特定的 Fuse pod 打开它，以便您可以监控正在运行的 Fuse 容器。

您从命令行部署 Fuse 控制台模板。



注意

要在 Minishift 或 CDK based environments 上安装 Fuse 控制台，请按照以下 KCS 文章中所述的步骤操作。

- 要在 Minishift 或 CDK based environments 上安装 Fuse 控制台，请参阅 [KCS 4998441](#)。
- 如果需要禁用 Jolokia 身份验证，请参阅 [KCS 3988671](#) 中描述的临时解决方案。

前提条件

- 如 OpenShift 的 Fuse 指南中所述，在 OpenShift 镜像流和 Fuse 控制台安装 Fuse。



注意

- **Fuse 控制台的用户管理由 OpenShift 处理。**
- **基于角色的访问控制（在部署后访问 Fuse 控制台的用户）还不适用于 OpenShift 3.11 上的 Fuse。**

第 3.1 节 “在 OpenShift 3.11 上部署 Fuse 控制台”

第 3.2 节 “从 OpenShift 3.11 上的 Fuse 控制台监控单个 Fuse pod”

3.1. 在 OPENSIFT 3.11 上部署 FUSE 控制台

表 3.1 “Fuse 控制台模板” 描述可用于从命令行部署 Fuse 控制台的 OpenShift 3.11 模板，具体取决于 Fuse 应用程序部署的类型。

表 3.1. Fuse 控制台模板

类型	描述
fis-console-cluster-template.json	Fuse 控制台可以发现并连接到在多个命名空间或项目中部署的 Fuse 应用程序。要部署此模板，您必须具有 OpenShift cluster-admin 角色。
fis-console-namespace-template.json	此模板限制 Fuse 控制台访问当前 OpenShift 项目（命名空间），因此充当单个租户部署。若要部署此模板，您必须拥有当前 OpenShift 项目的 admin 角色。

另外，您可以通过运行以下命令来查看所有模板的参数列表：

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fis-console-namespace-template.json
```



注意

Fuse 控制台模板默认配置端到端加密，以便您的 Fuse 控制台请求是从浏览器到集群服务的安全端到端。

前提条件

- 对于 OpenShift 3.11 上的集群模式，您需要集群管理员角色和集群模式模板。运行以下命令：

```
oc adm policy add-cluster-role-to-user cluster-admin system:serviceaccount:openshift-infra:template-instance-controller
```

流程

从命令行部署 Fuse 控制台：

1. 运行以下命令，基于 Fuse 控制台模板创建一个新应用程序（其中 **myproject** 是项目的名称）：

- 对于 **Fuse Console** 集群模板，其中 **myhost** 是用于访问 **Fuse** 控制台的主机名：

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fis-console-cluster-template.json -p ROUTE_HOSTNAME=myhost
```

- 对于 **Fuse Console** 命名空间模板：

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-7_13_0-00014-redhat-00001/fis-console-namespace-template.json
```



注意

您可以省略命名空间模板的 `route_hostname` 参数，因为 OpenShift 会自动生成一个。

2. 运行以下命令，获取 Fuse 控制台部署的状态和 URL：

```
oc status
```

3. 要从浏览器访问 **Fuse** 控制台，请使用提供的 URL。

Example:

+<https://fuse-console.192.168.64.12.nip.io>.

3.2. 从 OPENSIFT 3.11 上的 FUSE 控制台监控单个 FUSE POD

您可以为在 OpenShift 3.11 上运行的 Fuse pod 打开 Fuse 控制台。

前提条件

- 要将 OpenShift 配置为在 pod 视图中显示到 Fuse Console 的链接，在 OpenShift 镜像上运行 Fuse 的 pod 必须在 name 属性中声明 TCP 端口，设置为 jolokia：

```
{
  "kind": "Pod",
  [...]
  "spec": {
    "containers": [
      {
        [...]
        "ports": [
          {
            "name": "jolokia",
            "containerPort": 8778,
            "protocol": "TCP"
          }
        ]
      }
    ]
  }
}
```

流程

1. 在 OpenShift 项目中的 Applications → Pods 视图中，点 pod 名称查看正在运行的 Fuse pod 的详情。在此页面的右侧，您会看到容器模板的摘要：

Template

Containers

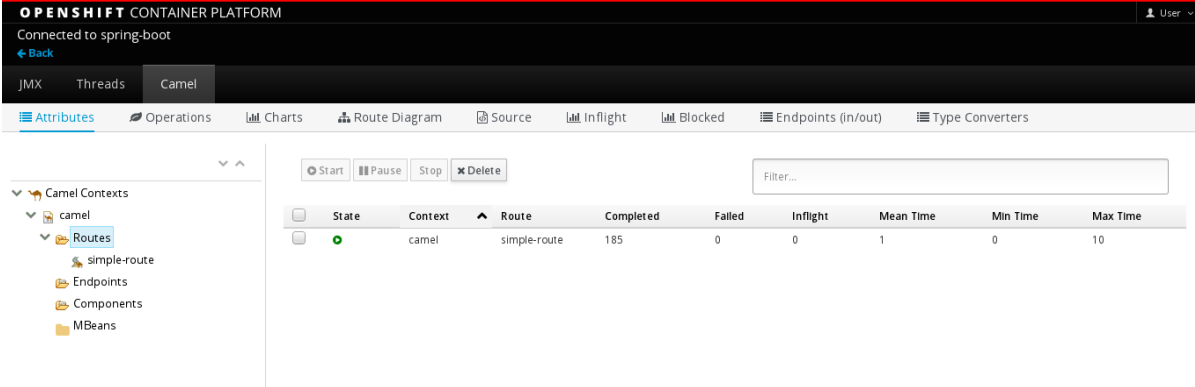
CONTAINER: SPRING-BOOT

-  **Image:** [test/fuse70-spring-boot](#) eda527f 193.1 MiB
-  **Build:** [fuse70-spring-boot-s2i, #2](#)
-  **Source:** Binary
-  **Ports:** 8080/TCP (http), 8778/TCP (jolokia), 9779/TCP (prometheus)
-  **Mount:** default-token-p4zsn → /var/run/secrets/kubernetes.io/serviceaccount
read-only
-  **CPU:** 200 millicores to 1 core
-  **Readiness Probe:** GET /health on port 8081 (HTTP) 10s delay, 1s timeout
-  **Liveness Probe:** GET /health on port 8081 (HTTP) 180s delay, 1s timeout

 [Open Java Console](#)

2.

从此视图中，单击 **Open Java Console** 链接以打开 Fuse 控制台。



State	Context	Route	Completed	Failed	Inflight	Mean Time	Min Time	Max Time
<input type="checkbox"/>	camel	simple-route	185	0	0	1	0	10

第 4 章 查看容器和应用程序

当您登录到 OpenShift 的 Fuse 控制台时，Fuse Console 主页会显示可用的容器。

流程

- 若要管理（创建、编辑或删除）容器，请使用 OpenShift 控制台。
- 要查看 OpenShift 集群上的 Fuse 应用程序和 AMQ Broker（如果适用），请点击 **Online** 选项卡。

第 5 章 查看并管理 APACHE CAMEL 应用程序

在 Fuse Console 的 Camel 选项卡中，您可以查看和管理 Apache Camel 上下文、路由和依赖项。

您可以查看以下详情：

- 所有正在运行的 Camel 上下文列表
- 每个 Camel 上下文（如 Camel 版本号和运行时静态）的详细信息
- 每个 Camel 应用程序及其运行时统计中的所有路由列表
- 正在运行的路由的图形表示以及实时指标

您还可以通过以下方法与 Camel 应用程序交互：

- 启动和暂停上下文
- 管理所有 Camel 应用程序及其路由的生命周期，以便您可以重启、停止、暂停、恢复等。
- 运行路由的实时追踪和调试
- 浏览消息并将其发送到 Camel 端点

前提条件

只有在连接到使用一个或多个 Camel 路由的容器时，Camel 选项卡才可用。

5.1. 启动、暂停或删除上下文

1. 在 **Camel** 选项卡的树视图中，单击 **Camel Contexts**。
2. 选中列表中一个或多个上下文旁边的框。
3. 点 **Start** 或 **Suspend**。
4. 删除上下文：
 - a. 停止上下文。
 - b. 单击 **ellipse** 图标，然后从下拉菜单中选择 **Delete**。



注意

当您删除上下文时，您可以从部署的应用程序中删除它。

5.2. 查看 CAMEL 应用程序详情

1. 在 **Camel** 选项卡的树视图中，点 **Camel** 应用程序。
2. 要查看应用属性和值的列表，请单击 **Attributes**。
3. 要查看应用程序属性的图形表示，请点 **Chart**，然后点 **Edit** 以选择要在图表中看到的属性。
4. 要查看动态和被阻止交换，请点击 **Exchanges**。
5. 要查看应用端点，请单击 **Endpoints**。您可以根据 **URL**、**Route ID** 和 **方向** 来过滤列表。
6. 要查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，用于将消息正文和消息标头转换为不同的类型，请单击 **Type Converters**。

7. 要查看和执行 JMX 操作，如从 XML 添加或更新路由或查找 classpath 中的所有 Camel 组件，请点 **Operations**。

5.3. 查看 CAMEL 路由列表并与之交互

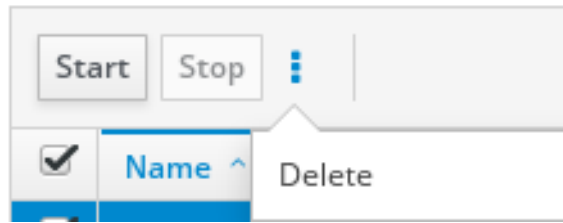
1. 查看路由列表：
 - a. 点 **Camel** 选项卡。
 - b. 在树视图中，点应用程序的路由文件夹：

Routes

<input type="button" value="Start"/> <input type="button" value="Stop"/> ⋮		
<input type="checkbox"/>	Name ^	State
<input type="checkbox"/>	_route1	Started
<input type="checkbox"/>	_route2	Started

2. 启动、停止或删除一个或多个路由：
 - a. 选中列表中一个或多个路由旁边的框。
 - b. 点 **Start** 或 **Stop**。
 - c. 要删除路由，您必须首先停止它。然后，单击 **ellipse** 图标，然后从下拉菜单中选择 **删除**。

Routes



注意

- 删除路由时，您可以将其从部署的应用程序中删除。
- 您还可以在树视图中选择特定的路由，然后单击右上角的菜单来启动、停止或删除它。

3.

要查看路由的图形图，请点 **Route** 图表。

4.

要查看动态和被阻止交换，请点击 **Exchanges**。

5.

要查看端点，请点 **Endpoints**。您可以根据 **URL**、**Route ID** 和方向过滤列表。

6.

点 **Type Converters** 查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，该机制用于将消息正文和消息标头转换为不同的类型。

7.

与特定路由交互：

a.

在 **Camel** 选项卡的树视图中，选择一个路由。

b.

要查看路由属性和值的列表，请单击 **Attributes**。

c.

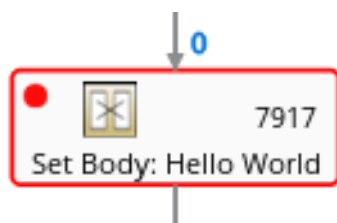
要查看路由属性的图形表示，请点 **Chart**。您可以点 **Edit** 来选择要在图表中看到的属性。

- d. 要查看动态和被阻止交换，请点击 **Exchanges**。
 - e. 点 **Operations** 查看并在路由上执行 **JMX** 操作，如将路由转储为 **XML** 或获取路由的 **Camel ID** 值。
8. 通过路由追踪信息：
- a. 在 **Camel** 选项卡的树视图中，选择一个路由。
 - b. 选择 **Trace**，然后单击 **Start tracing**。
9. 将信息发送到路由：
- a. 在 **Camel** 选项卡的树视图中，打开上下文的端点文件夹，然后选择端点。
 - b. 单击 **Send** 子选项卡。
 - c. 以 **JSON** 或 **XML** 格式配置消息。
 - d. 单击 **Send**。
 - e. 返回到路由的 **Trace** 选项卡，以查看消息通过路由流。

5.4. 调试路由

1. 在 **Camel** 选项卡的树视图中，选择一个路由。
2. 选择 **Debug**，然后单击 **Start debugging**。
3. 要添加断点，请在图中选择一个节点，然后单击 **Add breakpoint**。节点中会出现一个红色的

点：



节点添加到断点列表中：

Breakpoints	
setBody1	×
log1	×

4. 单击向下箭头，以进入下一节点或 **Play** 按钮，以恢复运行路由。
5. 单击 **暂停** 按钮，以暂停路由的所有线程。
6. 完成后点 **Stop debug**。所有断点都被清除。

第 6 章 查看 AMQ BROKER

您可以配置 Fuse 控制台来查看 OpenShift 集群中部署的所有 AMQ 代理。

先决条件

每个 AMQ 代理镜像（您要在 Fuse Console 中查看）都必须：

- 安装在同一 OpenShift 集群上，Fuse 控制台已安装在其中。
- 配置以便 Fuse 控制台可以识别和连接它，如 AMQ Broker 文档中的 [Fuse Console 中启用 Artemis 插件的章节中所述](#)。

流程

- 点 Artemis 查看 AMQ 管理控制台并监控 AMQ Broker 的状态。（AMQ Broker 基于 [Apache ActiveMQ Artemis](#)。）

有关使用 AMQ 管理控制台的详情，请参考 [管理 AMQ Broker 指南中的使用 AMQ 管理控制台](#)。

第 7 章 查看并管理 JMX 域和 MBEANS

Java 管理扩展(JMX)是一种 Java 技术，允许您在运行时动态管理资源（服务、设备和应用程序）。资源由名为 **MBeans**（用于 **Managed Bean**）的对象表示。您可以在资源创建、实施或安装后立即管理和监控资源。

通过 **Fuse Console** 上的 **JMX** 插件，您可以查看和管理 **JMX** 域和 **MBeans**。您可以查看 **MBean** 属性、运行命令和创建显示 **MBeans** 统计信息的图表。

JMX 选项卡提供活动 **JMX** 域和以文件夹组织的 **MBeans** 的树状视图。您可以在 **MBeans** 上查看详情和执行命令。

流程

1. 查看并编辑 **MBean** 属性：
 - a. 在树视图中，选择一个 **MBean**。
 - b. 单击 **Attributes** 选项卡。
 - c. 点一个属性查看其详情。
2. 执行操作：
 - a. 在树视图中，选择一个 **MBean**。
 - b. 单击 **Operations** 选项卡，展开列出的操作之一。
 - c. 单击 **Execute** 以运行该操作。
3. 查看图表：

- a. 在树视图中，选择一个项目。

- b. 点 **Chart** 选项卡。

第 8 章 查看并管理 QUARTZ SCHEDULES

quartz (<http://www.quartz-scheduler.org/>)是一个功能丰富的开源作业调度库，您可以在大多数 Java 应用程序内集成。您可以使用 Quartz 为执行作业创建简单或复杂的计划。作业被定义为一个标准 Java 组件，您可以几乎执行该组件来对其进行编程。

如果您的 Camel 路由部署了 camel-quartz2 组件，Fuse 控制台会显示 Quartz 选项卡。请注意，您还可以通过 JMX 树视图访问 Quartz mbeans。

流程

1.

在 Fuse 控制台中，单击 Quartz 选项卡。

Quartz 页面包含 Quartz Schedulers 和 Scheduler、Triggers 和 Jobs 选项卡的树状视图。

2.

要暂停或启动调度程序，请单击 调度程序选项卡上的按钮。

3.

点 Triggers 选项卡查看决定作业何时运行的触发器。例如，触发器可以指定在一天（到 millisecond）、在指定天数或指定次数或特定时间重复启动作业。

-

要过滤触发器列表，请从下拉列表中选择 State、Group、Name 或 Type。然后，您可以通过选择或键入 fill-on 字段来进一步过滤列表。

-

要暂停、恢复、更新或手动触发触发器，请单击 Action 列中的选项。

4.

点 Jobs 选项卡查看正在运行的作业列表。您可以根据表中的列对列表进行排序：Group、Name、Durable、Recover、Job ClassName 和 Description。

第 9 章 查看诊断

使用 **Diagnostics** 选项卡，通过 **JVM 诊断** 和 **HotspotDiagnostic** 接口查看有关 **JVM** 的诊断信息。



注意

这个功能与 **Java Mission Control (jmc)** 或命令行工具 **jcmd** 中的 **Diagnostic Commands** 视图类似。在某些情况下，该插件将提供对应的 **jcmd** 命令。

流程

1. 要检索已加载类的实例数量以及它们所消耗的字节数，请单击 **Class Histogram**。如果操作重复，选项卡会显示自上次运行以来的区别。
2. 要查看 **JVM 诊断** 标志设置，请单击 **JVM** 标志。
3. 对于正在运行的 **JVM**，您还可以修改标志设置。

其他资源

支持的 **JVM** 依赖于平台，如需更多信息，请访问以下源之一：

- <http://www.oracle.com/technetwork/java/vmoptions-jsp-140102.html>
- <http://openjdk.java.net/groups/hotspot/docs/RuntimeOverview.html>

第 10 章 查看线程

您可以查看和监控线程状态。

流程

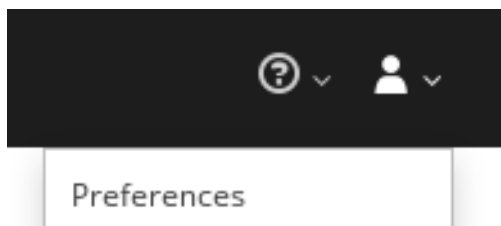
1. 单击 **Runtime** 选项卡，然后单击 **Threads** 子选项卡。**Threads** 页面列出了每个线程的活跃线程和堆栈追踪详情。默认情况下，线程列表以降序 ID 顺序显示所有线程。
2. 要通过增加 ID 进行排序，请点 ID 列标签。
3. (可选) 根据线程状态 (如 **Blocked**) 或线程名称过滤列表。
4. 要深入查看特定线程的详细信息，如锁定类名称和该线程的完整堆栈追踪，请在 **Actions** 列中点击 **More**。

第 11 章 确保数据在 FUSE 控制台中正确显示

如果在 Fuse 控制台中显示队列和连接缺少队列、缺少连接或显示不一致的图标，请调整 Jolokia 集合大小参数，用于指定响应中 Jolokia marshals 中的最大元素数。

流程

1. 在 Fuse 控制台右上角，单击用户图标，然后单击 **Preferences**。



2. 增加 **Maximum collection size** 选项的值（默认值为 50,000）。
3. 单击 **Close**。

附录 A. FUSE 控制台配置属性

默认情况下，Fuse Console 配置在 `hawtconfig.json` 文件中定义。您可以自定义 Fuse 控制台配置信息，如标题、徽标和登录页面信息。

表 A.1 “Fuse 控制台配置属性” 提供属性的描述，并列出了每个属性是否需要值。

表 A.1. Fuse 控制台配置属性

部分	属性名称	默认值	描述	必需？
关于	标题	Red Hat Fuse Management Console	在 Fuse 控制台的 About 页面中显示的标题。	必填
	productInfo	空值	在 Fuse 控制台的 About 页面中显示的产品信息。	选填
	additionalInfo	空值	在 Fuse 控制台的 About 页面中显示的任何其他信息。	选填
	版权	空值	在 Fuse 控制台 About 页面中显示的版权信息。	选填
	imgSrc	img/Logo-RedHat-A-Reverse- RGB.png	在 Fuse 控制台的 About 页面中出现的镜像。	必填
品牌	appName	Red Hat Fuse Management Console	应用程序的名称。此名称显示在 Fuse 控制台的标题栏中。	必填
	appLogoUrl	img/Logo-Red_Hat-Fuse- A-Reverse- RGB.png	在 Fuse Console <code>navigation</code> 栏中显示的应用程序徽标镜像文件的路径。该值可以是相对于 <code>Hawtio</code> 状态 URL 或绝对 URL 的路径。	必填

