Red Hat Fuse 7.8

迁移指南

迁移到 Red Hat Fuse 7.8

Last Updated: 2024-10-28

Red Hat Fuse 7.8 迁移指南

迁移到 Red Hat Fuse 7.8

法律通告

摘要

使用本指南帮助您将 Fuse 安装升级到最新版本的红帽 Fuse。

目录

前言

本指南提供有关更新红帽 Fuse 和 Fuse 应用程序的信息：

注意

如果要从 Fuse 6 迁移到最新的 Fuse 7 版本，然后再按照本指南中的说明，按照 Red Hat Fuse 7.0 迁移 指南中的说明进行操作。

第 1 章 升级 FUSE 在线

Fuse Online 升级过程取决于是否在 Red Hat OpenShift Online 或 OpenShift Container Platform (OCP) 上安装 Fuse Online。

- **OpenShift Online** - 当 Fuse 7.8 被发布时，OpenShift Online 上的 Fuse Online 基础架构会自动升级。您必须重新发布任何正在运行的集成，如 升级在 OpenShift Online 上运行的 Fuse Online 集成 中所述。
- **OCP** - 要升级在 OpenShift Container Platform on-site 上运行的 Fuse Online 环境，您应该首先为 Fuse 备份和恢复数据库设置 Docker 镜像的访问权限，如 升级前访问 Docker 镜像 中所述。升级 Fuse 的流程取决于您如何安装 Fuse：
  - 如果您使用 OperatorHub 安装 Fuse，请参阅使用 OperatorHub 升级 Fuse。
  - 如果您使用安装脚本安装 Fuse，请参阅使用 安装脚本 升级 Fuse。

  要升级集成，您应该考虑 Apache Camel 更新，如 Camel 迁移注意事项 中所述。

1.1. 在升级前访问 DOCKER 镜像

默认情况下，Fuse Online 升级过程从 **docker.io** registry 中拉取 Fuse Online 备份和恢复数据库的 Docker 镜像，而不是从 **registry.redhat.io** 中拉取（如已知问题 https://issues.redhat.com/browse/ENTESB-15364中所述）。因为 **docker.io** 对免费下载造成服务限制，所以 Fuse 在线升级过程在尝试下载备份和恢复数据库镜像时可能会遇到以下错误：

```
error: code = Unknown desc = toomanyrequests: You have reached your pull rate limit
```

有关 Docker Hub 拉取速率限制的更多信息，请参阅 Docker 文档(https://www.docker.com/increase-rate-limits)。

为了避免遇到 Docker 限制错误，请在启动 Fuse Online 升级过程前，请确保您可以访问 Docker 镜像。如何访问 Docker 镜像取决于您的 OpenShift Container Platform (OCP)版本：

- 对于 OCP 4.6，将 Docker Hub 凭证添加到现有的 syndesis pull secret 中，如 在 OCP 4.6 上访问 Docker 镜像 中所述。当您将 Docker 凭证添加到 syndesis pull secret 时，会引发 API 上的 Docker Hub 限制，具体取决于 Docker Hub 帐户类型。
- 对于 OCP 3.11，拉取 Fuse Online 备份和恢复数据库镜像，从 **docker.io** 恢复到本地缓存，如 在 OCP 3.11 上访问 Docker 镜像 中所述。

1.1.1. 访问 OCP4 上的 Docker 镜像

为了避免在启动 Fuse Online 升级过程前遇到可能的 Docker 限制错误，请将 docker 凭证添加到 syndesis pull secret 中。

前提条件

有 OpenShift 集群管理员访问权限。

流程

将 Docker Hub 凭证添加到现有 syndesis pull secret 中：

1. 使用您的管理员凭证登录到 OpenShift：

```
oc login -u system:admin
```

2. 运行以下命令检索您的 syndesis pull secret：

```
 oc get secrets syndesis-pull-secret -o=custom-columns=SECRET:.data.* --no-headers |
base64 -d | jq
```

这个命令返回类似如下的输出，其中 <AUTH> 是您在 base64 中编码的用户名和密码：

```
{
  "auths":{
    "registry.redhat.io":{
      "username":"<AUTH>",
      "password":"<AUTH>",
      "auth":"<AUTH>"
    },
    <You can have more auths elements here>
  }
 }
```

3. 要添加 Docker Hub registry 凭证，请编辑 pull secret，并将其保存到 PULL_SECRET 变量中：

```
 PULL_SECRET=$(base64 -w 0 <<EOF
{
  "auths":{
    "registry.redhat.io":{
      "username":"<AUTH>",
      "password":"<AUTH>",
      "auth":"<AUTH>"
    },
    <You can have more auths elements here>
    },
    "https://index.docker.io/v1/": {
      "auth": "<AUTH>"
    }
  }
}
EOF
)
```

4. 要修补您的 syndesis pull secret，请运行以下命令：

```
oc apply -f - <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: syndesis-pull-secret
data:
  .dockerconfigjson: $PULL_SECRET
type: kubernetes.io/dockerconfigjson
EOF
```

后续步骤

- 如果您使用 OperatorHub 安装 Fuse 在线，请参阅使用 OperatorHub 升级 Fuse Online。
- 如果您使用安装脚本在线安装 Fuse，请参阅使用 安装脚本 升级 Fuse Online。

1.1.2. 访问 OCP 3.11 上的 Docker 镜像

为了避免在 OCP 3.11 上启动 Fuse Online 升级过程前遇到可能的 Docker 限制错误，您应该将所需的镜像从 **docker.io** 拉取到本地缓存。

前提条件

有 OpenShift 集群管理员访问权限。

流程

1. 使用您的 OpenShift 用户名和 OpenShift 集群 URL 登录 OpenShift 节点：

```
ssh login <user>@node1.<clusterUrl>
```

2. 使用您的用户名和密码登录到 docker ：

```
sudo docker login -u <username> -p <password> docker.io
```

3. 运行以下命令拉取 PostgreSQL 镜像：

```
sudo docker pull crunchydata/crunchy-pgdump:centos7-10.11-4.2.1
sudo docker pull crunchydata/crunchy-pgrestore:centos7-10.11-4.2.1
sudo docker pull centos:7
```

   PostgresQL 镜像拉取到 OpenShift 节点的内部 Docker 注册表。

4. 为每个 OpenShift 节点重复前面的三个步骤。

后续步骤

按照使用安装脚本 升级 Fuse Online 中的步骤，升级 Fuse 在线。

1.2. 使用 OPERATORHUB 升级 FUSE 在线

从 OCP 4.6 开始，OperatorHub 中提供了 Fuse Online 7.8。如果使用 OCP 4.5，如果要安装 Fuse Online 7.8，则必须升级到 OCP 4.6。

注： 如果您使用安装脚本安装 Fuse Online 7.7，您应该使用安装脚本升级到 Fuse Online 7.8，如使用 安装脚本升级 Fuse Online 中所述。

安装 Fuse Online 时，您可以使用 **fuse-online-v7.n** 格式指定一个频道，其中 **n** 是当前的发行号。例如，对于 Fuse Online 7.8，频道为 **fuse-online-v7.8**。

从 Fuse Online 7.8 版本升级到较新的 Fuse Online 7.8 版本取决于您在安装 Fuse Online 时选择 **的批准策略** ：

- 对于自动更新，当有新版本的 Fuse Online operator 可用时，OpenShift Operator Lifecycle Manager (OLM)将自动升级 Fuse Online 的运行实例，而无需人为干预。

- 对于 手动更新，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，才能将 Fuse Online operator 更新至新版本，如 OpenShift 文档中的 手动批准待处理的 Operator 升级 部分所述。

在基础架构升级过程中和之后，现有集成将继续与 Fuse Online 库和依赖项 *旧版本* 运行。要让它们使用更新版本运行，您必须重新发布它们。

对于 OCP 4.6 及更新的版本，要从 Fuse Online 7.7 升级到 7.8，请使用以下步骤。

流程

1. 为了避免遇到 Docker 限制错误，请在启动 Fuse Online 升级过程前，将 docker 凭证添加到 syndesis pull secret 中，如 在升级前访问 Docker 镜像 中所述。
2. 升级 Fuse Online 操作器：
   a. 在 OpenShift Web 控制台中，点 Operators >Installed Operators。
   b. 单击 Fuse Online 操作器，然后单击 Subscription。
   c. 在 频道旁边 点击 Edit 图标。
   d. 选择 fuse-online-v7.8 频道，然后单击 Save。

如果在安装 Fuse Online 时指定了 Manual 更新，请按照 OpenShift 文档中的手动批准待处理的 Operator 升级部分中的说明批准 Operator 更新请求。

1.3. 使用安装脚本升级 FUSE ONLINE

如果您使用安装脚本（而不是 OperatorHub）安装 Fuse Online，以下是升级 Fuse 在线的一般步骤：

- 集群管理员为 Fuse Online 备份和恢复数据库设置对 Docker 镜像的访问权限：
  - 对于 OCP 3.11，拉取 Fuse Online 备份和恢复数据库镜像，从 docker.io 恢复到本地缓存。
  - 对于 OCP 4.6，将 Docker Hub 凭证添加到现有的 syndesis pull secret 中。
- 下载最新的 Fuse Online 版本。
- 获取从集群管理员升级 Fuse Online 的权限。
- 运行更新脚本。

以下升级的升级步骤是相同的：

- 从 Fuse Online 7.7 到 Fuse Online 7.8
- 从 Fuse Online 7.8 版本到较新的 Fuse Online 7.8 版本

先决条件

- 您已在 OCP 现场安装并正在运行 Fuse Online 版本 7.7。或者，您在 OCP 3.11 上安装并正在运行 7.8 of Fuse Online 版本，并希望升级到新的应用程序镜像。
  对于早期版本：
  - 如果您在 OCP 上运行 Fuse Online 版本 7.6，则必须 升级到 7.7，然后升级到 7.8。

- 如果您在 OCP 上运行 Fuse Online 版本 7.5，则必须 升级到 7.6，然后升级到 7.7。
- 如果您在 OCP 上运行 Fuse Online 版本 7.4，则必须 升级到 7.5，然后升级到 7.6。
- 如果您在 OCP 上运行 Fuse Online 版本 7.3，则必须 升级到 7.4，然后您可以升级到 7.5。
- 如果您在 OCP 上运行 Fuse Online 版本 7.2，则必须 升级到 7.3。
- 如果您在 OCP 上运行 Fuse Online 版本 7.1，则必须 升级到 7.2。

- 已安装 oc 客户端工具，并连接到安装 Fuse Online 的 OCP 集群。
- 您有集群管理权限，这是此流程中前两个步骤所需的。

流程

1. 为避免遇到可能的 Docker 限制错误，集群管理员会设置对 Docker 镜像的访问权限，如 在升级前访问 Docker 镜像 中所述。
2. 集群管理员下载 Fuse Online 软件包，并授予用户在特定项目中升级 Fuse Online 的权限：
   a. 从以下位置下载包含 Fuse 在线安装脚本的软件包：
   https://github.com/syndesisio/fuse-online-install/releases/tag/1.11

   在文件系统的便捷位置解包下载的存档。fuse-online-install-1.11 目录包含用于升级 Fuse Online 的脚本和支持文件。

   b. 更改到包含提取的存档的目录。例如：
   cd fuse-online-install-1.11

   c. 使用集群管理帐户登录到 OpenShift，例如：
   oc login -u admin -p admin

   d. 切换到需要升级 Fuse Online 的 OpenShift 项目，例如：
   oc project fuse-online-project

   e. 更新 Fuse Online 自定义资源定义：
   bash install_ocp.sh --setup

   f. 仅授予在此项目中升级 Fuse Online 的权限。例如，以下命令向 developer 用户授予将 Fuse Online 升级权限。集群管理员运行此命令后，开发人员用户可以在此项目中升级 Fuse Online，本例中为 fuse-online-project ：

   bash install_ocp.sh --grant developer

3. 授予升级 Fuse Online 的权限的用户执行升级：

   a. 登录到 OpenShift，例如：

   oc login -u developer

b. 切换到要升级 Fuse Online 的项目，例如：

```
oc project fuse-online-project
```

c. 要检查您要升级到的版本，请使用-- version 选项运行更新脚本，如下所示：

```
bash update_ocp.sh --version
```

d. 按如下方式调用更新脚本：

```
bash update_ocp.sh
```

要了解更多有关脚本的信息，请调用 bash update_ocp.sh --help。

在基础架构升级过程中和之后，现有集成将继续与 Fuse Online 库和依赖项 *旧版本* 运行。

4. 升级运行的 Fuse Online 集成，如下所示：

a. 在 Fuse Online 中，选择您要升级的集成。

b. 选择 编辑。

c. 选择 Publish 重新发布集成。

重新发布集成会强制重新构建使用最新的 Fuse Online 依赖项。

1.4. CAMEL 迁移注意事项

Red Hat Fuse 使用 Apache Camel 2.23。升级到 Fuse 7.8 时，您应该考虑对 Camel 2.22 和 2.23 的以下更新。

Camel 2.22 更新

- Camel 已从 Spring Boot v1 升级到 v2，因此不再支持 v1。
- 升级到 Spring Framework 5。Camel 应该与 Spring 4.3.x 一起工作，但转发 Spring 5.x 将在以后的版本中是 Spring 5.x 的最低 Spring 版本。
- 升级到 Karaf 4.2。您可以在 Karaf 4.1 上运行 Camel，但在此发行版本中，我们仅官方支持 Karaf 4.2。
- 使用 toD DSL 优化，为可能出现的组件重复使用端点和制作者。例如，基于 HTTP 的组件将重复使用发送到同一主机的动态 URI 的生成者(HTTP 客户端)。
- 现在，具有 read-lock idempotent/idempotent-changed 的 File2 使用者可以被配置为延迟发行任务，以便在处理过程中使用文件扩展窗口，该文件可以在带有共享幂等存储库的主动/主动集群设置中可用，以确保其他节点无法快速将已处理的文件视为可处理的文件（仅当您具有 readLockRemoveOnCommit=true 时）。
- 允许在 request/reply 模式中在 Netty4 producer 上插件自定义请求/回复关联 ID 管理器。author 组件现在默认使用扩展模式来支持大于 140 个字符的 12 月
- REST DSL 生成者现在支持使用 endpointProperties 在 REST 配置中配置。
- Kafka 组件现在支持 HeaderFilterStrategy 插件自定义实现，以控制 Camel 和 Kafka 信息之间的标头映射。
- REST DSL 现在支持客户端请求验证，以验证 Content-Type/Accept 标头是否可以用于 REST 服务。
- Camel 现在有一个 Service Registry SPI，它允许您使用 Camel 实现或 Spring Cloud 将路由注册到服务 registry （如 consul、etcd 或 zookeeper）。
- SEDA 组件现在的默认队列大小为 1000，而不是无限。

- 解决了以下值得注意的问题：
  - 修复了 camel-cxf consumer 的 CXF continuation 超时问题，可能会导致消费者返回带有数据的响应，而不是向调用 SOAP 客户端触发超时。
  - 修复了 camel-cxf consumer 在使用强大的单向操作时，不会发布 UoW。
  - 修复了使用 AdviceWith，并使用 weave onException 等方法无法正常工作。
  - 修复了并行处理和流模式中的 Splitter 可能会阻断，而当迭代消息正文（）ator 在第一个调用的 next （）方法调用中抛出异常时。
  - 修复了 Kafka 使用者，如果 autoCommitEnable=false，则不会自动提交。
  - 修复了文件消费者默认使用 markerFile 作为 read-lock，这应该为 none。
  - 修复了使用带有 Kafka 的手动提交以提供当前的记录偏移而不是之前（第一个为 -1 ）。
  - 修复了 Java DSL 中的基于内容路由器的问题可能无法解析 when predicates 中的属性占位符。

Camel 2.23 更新

- 升级到 Spring Boot 2.1。
- 现在，可以使用 spring-boot 自动配置来配置其他组件级选项。这些选项包含在 spring-boot 组件元数据 JSON 文件描述符中，以获得工具协助。
- 添加了一个文档部分，其中包含所有组件、数据格式和语言的所有 Spring Boot 自动配置选项。
-

现在，所有 Camel Spring Boot starter JARs 都会在其 JARs 中包含 META-INF/spring-autoconfigure-metadata.properties 文件，以优化 Spring Boot 自动配置。

- Throttler 现在支持基于动态表达式的关联组，以便您可以将消息分组到不同的节流集中。
- Hystrix EIP 现在允许 Camel 的错误处理程序继承，以便在您启用了带有红色错误处理时再次重试整个 Hystrix EIP 块。
- SQL 和 ElSql 用户现在支持路由格式的动态查询参数。请注意，这个功能仅限于使用简单表达式调用 Bean。
- swagger-restdsl maven 插件现在支持从 Swagger 规格文件中生成 DTO 模型类。
- 解决了以下值得注意的问题：
  - 已修复 Aggregator2，使其不会传播用于强制完成所有组的控制标头，因此当路由期间使用另一个聚合器 EIP 时不会再次发生。
  - 修复了错误处理程序中被重新传送的 Tracer 无法正常工作。
  - XML 文档的内置类型转换器可能会输出将错误解析到 stdout，现在已修复为使用日志记录 API 的输出。
  - 修复了使用 charset 选项（如果消息正文基于流传输）的 SFTP 编写文件无法正常工作。
  - 修复了当通过多个路由的路由时，不会重复使用 Zipkin root id，以便将它们分组到一个父范围中。
  - 修复了在主机名包含带有数字的 IP 地址时，使用 HTTP 端点时优化为D。
  - 修复了在通过临时队列进行请求/回复以及使用手动确认模式的 RabbitMQ 的问题。它不会确认临时队列（需要尽可能发出请求/回复）。

- 修复了在 Allow 标头中为 OPTIONS 请求返回所有允许的 HTTP 动词（如使用 rest-dsl 时）的各种 HTTP 消费者组件。

- 修复了 FluentProducerTemplate 的 thread-safety 问题。

第 2 章 升级到 SPRING BOOT 2

本章介绍了如何将应用程序从 Spring Boot 1 升级到 Spring Boot 2.0。

2.1. 开始前

在开始迁移到 Spring Boot 2 前，您必须查看系统要求和依赖项。

- 升级到最新的 1.5.x 版本
  - 在开始前，请升级到最新的 1.5.x 可用版本。这是为了确保您针对该行的最新依赖项进行构建。
- 检查依赖项
  - 迁移到 Spring Boot 2 将导致升级多个依赖项。查看 1.5.x 的依赖项管理以及 2.0.x 的依赖项管理，以评估您的项目如何受到影响。
  - 识别不是由 Spring Boot 管理的依赖项的兼容版本，然后为这些定义显式版本。
- 查看自定义配置
  - 可能需要在升级前检查项目定义的任何自定义配置。如果这可以通过使用标准自动配置来替换，请在升级前执行此操作。
- 查看系统要求
  - Spring Boot 2.0 需要 Java 8 或更高版本。
  - 它还需要 Spring Framework 5.0。

- Java 6 和 7 不再被支持。

2.2. 从 SPRING BOOT 1 升级到 SPRING BOOT 2

检查项目及其依赖项的状态后，升级到 Spring Boot 2.x 的最新维护版本。建议您在阶段升级。例如，首先从 Spring Boot 1.5 升级到 Spring Boot 2.0，然后升级到 2.1，然后升级到 Spring Boot 2 的最新维护版本。

迁移配置属性

在 Spring Boot 2.0 中，很多配置属性被重命名或删除。因此，您需要相应地更新 application.properties/application.yml。您可以使用新的 spring-boot-properties-migrator 模块来实现这一点。将依赖项作为依赖项添加到项目后，这不仅会分析应用的环境，并在启动时打印诊断，同时在您运行时临时为您迁移属性。

流程

1. 将 spring-boot-properties-migrator 模块添加到项目的 pom.xml 的 dependencies 部分。

```
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-properties-migrator</artifactId>
 <scope>runtime</scope>
</dependency>

runtime("org.springframework.boot:spring-boot-properties-migrator")
```

注意

完成迁移后，请确保从项目的依赖项中删除此模块。

第 3 章 在 SPRING BOOT 独立上升级 FUSE 应用程序

在 Spring Boot 上升级 Fuse 应用程序：

- 您应该考虑 Apache Camel 更新，如 第 3.1 节 “Camel 迁移注意事项” 所述。
- 您必须更新 Fuse 项目的 Maven 依赖项，以确保您使用正确的 Fuse 版本。

通常，您可以使用 Maven 来构建 Fuse 应用程序。Maven 是 Apache 的免费开源构建工具。Maven 配置在 Fuse 应用程序项目的 pom.xml 文件中定义。在构建 Fuse 项目时，Maven 会搜索外部存储库并下载所需的工件。您可以将 Fuse Bill of Materials (BOM)的依赖关系添加到 pom.xml 文件中，以便 Maven 构建过程会获取正确的 Fuse 支持的工件集合。

以下小节提供有关 Maven 依赖项的信息，以及如何在 Fuse 项目中更新它们。

- 第 3.2 节 “关于 Maven 依赖项”
- 第 3.3 节 “更新 Fuse 项目的 Maven 依赖项”

3.1. CAMEL 迁移注意事项

Red Hat Fuse 使用 Apache Camel 2.23。升级到 Fuse 7.8 时，您应该考虑对 Camel 2.22 和 2.23 的以下更新。

Camel 2.22 更新

- Camel 已从 Spring Boot v1 升级到 v2，因此不再支持 v1。
- 升级到 Spring Framework 5。Camel 应该与 Spring 4.3.x 一起工作，但转发 Spring 5.x 将在以后的版本中是 Spring 5.x 的最低 Spring 版本。
- 升级到 Karaf 4.2。您可以在 Karaf 4.1 上运行 Camel，但在此发行版本中，我们仅官方支持 Karaf 4.2。

- 使用 toD DSL 优化，为可能出现的组件重复使用端点和制作者。例如，基于 HTTP 的组件将重复使用发送到同一主机的动态 URI 的生成者(HTTP 客户端)。

- 现在，具有 read-lock idempotent/idempotent-changed 的 File2 使用者可以被配置为延迟发行任务，以便在处理过程中使用文件扩展窗口，该文件可以在带有共享幂等存储库的主动/主动集群设置中可用，以确保其他节点无法快速将已处理的文件视为可处理的文件（仅当您具有 readLockRemoveOnCommit=true 时）。

- 允许在 request/reply 模式中在 Netty4 producer 上插件自定义请求/回复关联 ID 管理器。author 组件现在默认使用扩展模式来支持大于 140 个字符的 12 月

- REST DSL 生成者现在支持使用 endpointProperties 在 REST 配置中配置。

- Kafka 组件现在支持 HeaderFilterStrategy 插件自定义实现，以控制 Camel 和 Kafka 信息之间的标头映射。

- REST DSL 现在支持客户端请求验证，以验证 Content-Type/Accept 标头是否可以用于 REST 服务。

- Camel 现在有一个 Service Registry SPI，它允许您使用 Camel 实现或 Spring Cloud 将路由注册到服务 registry （如 consul、etcd 或 zookeeper）。

- SEDA 组件现在的默认队列大小为 1000，而不是无限。

- 解决了以下值得注意的问题：

  - 修复了 camel-cxf consumer 的 CXF continuation 超时问题，可能会导致消费者返回带有数据的响应，而不是向调用 SOAP 客户端触发超时。

  - 修复了 camel-cxf consumer 在使用强大的单向操作时，不会发布 UoW。

  - 修复了使用 AdviceWith，并使用 weave onException 等方法无法正常工作。

- 修复了并行处理和流模式中的 Splitter 可能会阻断，而当迭代消息正文（）ator 在第一个调用的 next （）方法调用中抛出异常时。

- 修复了 Kafka 使用者，如果 autoCommitEnable=false，则不会自动提交。

- 修复了文件消费者默认使用 markerFile 作为 read-lock，这应该为 none。

- 修复了使用带有 Kafka 的手动提交以提供当前的记录偏移而不是之前（第一个为 -1 ）。

- 修复了 Java DSL 中的基于内容路由器的问题可能无法解析 when predicates 中的属性占位符。

Camel 2.23 更新

- 升级到 Spring Boot 2.1。

- 现在，可以使用 spring-boot 自动配置来配置其他组件级选项。这些选项包含在 spring-boot 组件元数据 JSON 文件描述符中，以获得工具协助。

- 添加了一个文档部分，其中包含所有组件、数据格式和语言的所有 Spring Boot 自动配置选项。

- 现在，所有 Camel Spring Boot starter JARs 都会在其 JARs 中包含 META-INF/spring-autoconfigure-metadata.properties 文件，以优化 Spring Boot 自动配置。

- Throttler 现在支持基于动态表达式的关联组，以便您可以将消息分组到不同的节流集中。

- Hystrix EIP 现在允许 Camel 的错误处理程序继承，以便在您启用了带有红色错误处理时再次重试整个 Hystrix EIP 块。

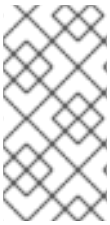
- SQL 和 ElSql 用户现在支持路由格式的动态查询参数。请注意，这个功能仅限于使用简单表达式调用 Bean。

- swagger-restdsl maven 插件现在支持从 Swagger 规格文件中生成 DTO 模型类。
- 解决了以下值得注意的问题：
  - 已修复 Aggregator2，使其不会传播用于强制完成所有组的控制标头，因此当路由期间使用另一个聚合器 EIP 时不会再次发生。
  - 修复了错误处理程序中被重新传送的 Tracer 无法正常工作。
  - XML 文档的内置类型转换器可能会输出将错误解析到 stdout，现在已修复为使用日志记录 API 的输出。
  - 修复了使用 charset 选项（如果消息正文基于流传输）的 SFTP 编写文件无法正常工作。
  - 修复了当通过多个路由的路由时，不会重复使用 Zipkin root id，以便将它们分组到一个父范围中。
  - 修复了在主机名包含带有数字的 IP 地址时，使用 HTTP 端点时优化为D。
  - 修复了在通过临时队列进行请求/回复以及使用手动确认模式的 RabbitMQ 的问题。它不会确认临时队列（需要尽可能发出请求/回复）。
  - 修复了在 Allow 标头中为 OPTIONS 请求返回所有允许的 HTTP 动词（如使用 rest-dsl 时）的各种 HTTP 消费者组件。
  - 修复了 FluentProducerTemplate 的 thread-safety 问题。

3.2. 关于 MAVEN 依赖项

Maven Bill of Materials (BOM) 文件的目的是提供一组策展的 Maven 依赖关系版本，它们可以正常工作，从而为您为每个 Maven 工件单独定义版本。

每个容器都有一个专用的 BOM 文件，在其中运行 Fuse。



注意

您可以在此处找到这些 BOM 文件 ：https://github.com/jboss-fuse/redhat-fuse。或者，请参阅 最新的发行注记 以了解有关 BOM 文件更新的信息。

Fuse BOM 提供以下优点：

- 定义 Maven 依赖项的版本，以便在将依赖项添加到 pom.xml 文件时不需要指定版本。
- 定义一组策展的依赖关系，它们被完全测试并支持特定版本的 Fuse。
- 简化 Fuse 的升级.

重要

红帽仅支持由 Fuse BOM 定义的依赖项集合。

3.3. 更新 FUSE 项目的 MAVEN 依赖项

要为 Spring Boot 升级 Fuse 应用程序，请更新项目的 Maven 依赖项。

流程

1. 打开项目的 pom.xml 文件。
2. 在项目的 pom.xml 文件中添加 dependencyManagement 元素（或者在父 pom.xml 文件中），如下例所示：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<project ...>
  ...
  <properties>
```

```
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <!-- configure the versions you want to use here -->
    <fuse.version>7.8.0.fuse-sb2-780038-redhat-00001</fuse.version>

  </properties>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.jboss.redhat-fuse</groupId>
        <artifactId>fuse-springboot-bom</artifactId>
        <version>${fuse.version}</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>
  ...
</project>
```

注意

确保也更新 Spring Boot 版本。这通常在 pom.xml 文件中的 Fuse 版本下找到：

```
  <properties>
    <!-- configure the versions you want to use here -->
    <fuse.version>7.8.0.fuse-sb2-780038-redhat-00001</fuse.version>
    <spring-boot.version>2.3.4.RELEASE</spring-boot.version>
  </properties>
```

3. **保存 pom.xml 文件。**

在 pom.xml 文件中将 BOM 指定为依赖项后，可以在 pom.xml 文件中添加 Maven 依赖项，*而无需* 指定工件版本。例如，要为 camel-velocity 组件添加依赖项，您要将以下 XML 片段添加到 pom.xml 文件中的 dependencies 元素中：

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-velocity</artifactId>
  <scope>provided</scope>
</dependency>
```

请注意，此依赖项定义中如何省略 version 元素。

第 4 章 在 JBOSS EAP 独立上升级 FUSE 应用程序

在 JBoss EAP 上升级 Fuse 应用程序：

- 您应该考虑 Apache Camel 更新，如 第 4.1 节 “Camel 迁移注意事项” 所述。
- 您必须更新 Fuse 项目的 Maven 依赖项，以确保您使用正确的 Fuse 版本。

  通常，您可以使用 Maven 来构建 Fuse 应用程序。Maven 是 Apache 的免费开源构建工具。Maven 配置在 Fuse 应用程序项目的 pom.xml 文件中定义。在构建 Fuse 项目时，Maven 会搜索外部存储库并下载所需的工件。您可以将 Fuse Bill of Materials (BOM)的依赖关系添加到 pom.xml 文件中，以便 Maven 构建过程会获取正确的 Fuse 支持的工件集合。

  以下小节提供有关 Maven 依赖项的信息，以及如何在 Fuse 项目中更新它们。

  - 第 4.2 节 “关于 Maven 依赖项”
  - 第 4.3 节 “更新 Fuse 项目的 Maven 依赖项”
- 您必须更新 Fuse 项目的 Maven 依赖项，以确保您使用 Java EE 依赖项的升级版本，如 第 4.4 节 “升级 Java EE 依赖项” 所述。

4.1. CAMEL 迁移注意事项

Red Hat Fuse 使用 Apache Camel 2.23。升级到 Fuse 7.8 时，您应该考虑对 Camel 2.22 和 2.23 的以下更新。

Camel 2.22 更新

- Camel 已从 Spring Boot v1 升级到 v2，因此不再支持 v1。
- 升级到 Spring Framework 5。Camel 应该与 Spring 4.3.x 一起工作，但转发 Spring 5.x 将在以后的版本中是 Spring 5.x 的最低 Spring 版本。

- 升级到 Karaf 4.2。您可以在 Karaf 4.1 上运行 Camel，但在此发行版本中，我们仅官方支持 Karaf 4.2。

- 使用 toD DSL 优化，为可能出现的组件重复使用端点和制作者。例如，基于 HTTP 的组件将重复使用发送到同一主机的动态 URI 的生成者(HTTP 客户端)。

- 现在，具有 read-lock idempotent/idempotent-changed 的 File2 使用者可以被配置为延迟发行任务，以便在处理过程中使用文件扩展窗口，该文件可以在带有共享幂等存储库的主动/主动集群设置中可用，以确保其他节点无法快速将已处理的文件视为可处理的文件（仅当您具有 readLockRemoveOnCommit=true 时）。

- 允许在 request/reply 模式中在 Netty4 producer 上插件自定义请求/回复关联 ID 管理器。author 组件现在默认使用扩展模式来支持大于 140 个字符的 12 月

- REST DSL 生成者现在支持使用 endpointProperties 在 REST 配置中配置。

- Kafka 组件现在支持 HeaderFilterStrategy 插件自定义实现，以控制 Camel 和 Kafka 信息之间的标头映射。

- REST DSL 现在支持客户端请求验证，以验证 Content-Type/Accept 标头是否可以用于 REST 服务。

- Camel 现在有一个 Service Registry SPI，它允许您使用 Camel 实现或 Spring Cloud 将路由注册到服务 registry （如 consul、etcd 或 zookeeper）。

- SEDA 组件现在的默认队列大小为 1000，而不是无限。

- 解决了以下值得注意的问题：

  - 修复了 camel-cxf consumer 的 CXF continuation 超时问题，可能会导致消费者返回带有数据的响应，而不是向调用 SOAP 客户端触发超时。

  - 修复了 camel-cxf consumer 在使用强大的单向操作时，不会发布 UoW。

- 修复了使用 AdviceWith，并使用 weave onException 等方法无法正常工作。
- 修复了并行处理和流模式中的 Splitter 可能会阻断，而当迭代消息正文（）ator 在第一个调用的 next （）方法调用中抛出异常时。
- 修复了 Kafka 使用者，如果 autoCommitEnable=false，则不会自动提交。
- 修复了文件消费者默认使用 markerFile 作为 read-lock，这应该为 none。
- 修复了使用带有 Kafka 的手动提交以提供当前的记录偏移而不是之前（第一个为 -1 ）。
- 修复了 Java DSL 中的基于内容路由器的问题可能无法解析 when predicates 中的属性占位符。

Camel 2.23 更新

- 升级到 Spring Boot 2.1。
- 现在，可以使用 spring-boot 自动配置来配置其他组件级选项。这些选项包含在 spring-boot 组件元数据 JSON 文件描述符中，以获得工具协助。
- 添加了一个文档部分，其中包含所有组件、数据格式和语言的所有 Spring Boot 自动配置选项。
- 现在，所有 Camel Spring Boot starter JARs 都会在其 JARs 中包含 META-INF/spring-autoconfigure-metadata.properties 文件，以优化 Spring Boot 自动配置。
- Throttler 现在支持基于动态表达式的关联组，以便您可以将消息分组到不同的节流集中。
- Hystrix EIP 现在允许 Camel 的错误处理程序继承，以便在您启用了带有红色错误处理时再次重试整个 Hystrix EIP 块。

- SQL 和 ElSql 用户现在支持路由格式的动态查询参数。请注意，这个功能仅限于使用简单表达式调用 Bean。

- swagger-restdsl maven 插件现在支持从 Swagger 规格文件中生成 DTO 模型类。

- 解决了以下值得注意的问题：

  - 已修复 Aggregator2，使其不会传播用于强制完成所有组的控制标头，因此当路由期间使用另一个聚合器 EIP 时不会再次发生。

  - 修复了错误处理程序中被重新传送的 Tracer 无法正常工作。

  - XML 文档的内置类型转换器可能会输出将错误解析到 stdout，现在已修复为使用日志记录 API 的输出。

  - 修复了使用 charset 选项（如果消息正文基于流传输）的 SFTP 编写文件无法正常工作。

  - 修复了当通过多个路由的路由时，不会重复使用 Zipkin root id，以便将它们分组到一个父范围中。

  - 修复了在主机名包含带有数字的 IP 地址时，使用 HTTP 端点时优化为D。

  - 修复了在通过临时队列进行请求/回复以及使用手动确认模式的 RabbitMQ 的问题。它不会确认临时队列（需要尽可能发出请求/回复）。

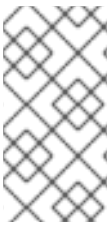
  - 修复了在 Allow 标头中为 OPTIONS 请求返回所有允许的 HTTP 动词（如使用 rest-dsl 时）的各种 HTTP 消费者组件。

  - 修复了 FluentProducerTemplate 的 thread-safety 问题。

4.2. 关于 MAVEN 依赖项

Maven Bill of Materials (BOM) 文件的目的是提供一组策展的 Maven 依赖关系版本，它们可以正常工作，从而为您为每个 Maven 工件单独定义版本。

每个容器都有一个专用的 BOM 文件，在其中运行 Fuse。



注意

您可以在此处找到这些 BOM 文件 ：https://github.com/jboss-fuse/redhat-fuse。或者，请参阅 最新的发行注记 以了解有关 BOM 文件更新的信息。

Fuse BOM 提供以下优点：

- 定义 Maven 依赖项的版本，以便在将依赖项添加到 pom.xml 文件时不需要指定版本。
- 定义一组策展的依赖关系，它们被完全测试并支持特定版本的 Fuse。
- 简化 Fuse 的升级.

重要

红帽仅支持由 Fuse BOM 定义的依赖项集合。

4.3. 更新 FUSE 项目的 MAVEN 依赖项

要为 JBoss EAP 升级 Fuse 应用程序，请更新项目的 Maven 依赖项。

流程

1. 打开项目的 pom.xml 文件。
2. 在项目的 pom.xml 文件中添加 dependencyManagement 元素（或者在父 pom.xml 文件中），如下例所示：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<project ...>
  ...
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <!-- configure the versions you want to use here -->
    <fuse.version>7.8.0.fuse-sb2-780038-redhat-00001</fuse.version>

  </properties>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.jboss.redhat-fuse</groupId>
        <artifactId>fuse-eap-bom</artifactId>
        <version>${fuse.version}</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>
  ...
</project>
```

3. 保存 pom.xml 文件。

在 pom.xml 文件中将 BOM 指定为依赖项后，可以在 pom.xml 文件中添加 Maven 依赖项，*而无需* 指定工件版本。例如，要为 camel-velocity 组件添加依赖项，您要将以下 XML 片段添加到 pom.xml 文件中的 dependencies 元素中：

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-velocity</artifactId>
  <scope>provided</scope>
</dependency>
```

请注意，此依赖项定义中如何省略 version 元素。

4.4. 升级 JAVA EE 依赖项

在 Fuse 7.8 中，BOM 文件中的一些受管依赖项已更新了 groupId 或 artifactId 属性，因此您必须相应地更新项目的 pom.xml 文件。

流程

1. 打开项目的 pom.xml 文件。

2. 要将 org.jboss.spec.javax.transaction 版本从 1.2 改为 1.3，将 org.jboss.spec.javax.servlet 版本从 3.1 改为 4.0，请更新以下示例所示的依赖项：

```
<dependency>
  <groupId>org.jboss.spec.javax.transaction</groupId>
  <artifactId>jboss-transaction-api_1.3_spec</artifactId>
</dependency>

<dependency>
    <groupId>org.jboss.spec.javax.servlet</groupId>
    <artifactId>jboss-servlet-api_4.0_spec</artifactId>
</dependency>
```

3. 要从 Java EE API 迁移到 Jakarta EE，请为每个 groupId 替换 javax 198.51.100.0/24，并修改单个依赖项的 artifactId，如下例所示：

```
<dependency>
  <groupId>jakarta.validation</groupId>
  <artifactId>jakarta.validation-api</artifactId>
</dependency>

<dependency>
  <groupId>jakarta.enterprise</groupId>
  <artifactId>jakarta.enterprise.cdi-api</artifactId>
</dependency>

<dependency>
  <groupId>jakarta.inject</groupId>
  <artifactId>jakarta.inject-api</artifactId>
</dependency>
```

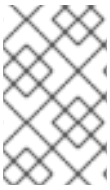
4.5. 在 JBOSS EAP 安装中升级现有的 FUSE

以下流程描述了如何升级 JBoss EAP 安装中的现有 Fuse。

流程

1. 要从一个 JBoss EAP 次版本升级到另一个版本，您应该遵循 JBoss EAP 补丁和升级指南 中的说明。

2. 要更新 Fuse，您必须在 JBoss EAP 安装程序上运行 Fuse，如在 JBoss EAP 上安装 指南 中所述。



注意

您不需要重新编译或重新采用 Fuse 应用程序。

4.6. 同时升级 FUSE 和 JBOSS EAP

以下流程描述了如何升级 Fuse 安装和 JBoss EAP 运行时，例如，如果您从 JBoss EAP 7.2 上的 Fuse 7.7 迁移到 JBoss EAP 7.3 上的 Fuse 7.8。

警告

在 同时升级 Fuse 和 JBoss EAP 运行时，红帽建议您执行 Fuse 和 JBoss EAP 运行时的全新安装。

流程

1. 要执行 JBoss EAP 运行时的新安装，请按照在 JBoss EAP 上安装 指南中的说明进行操作。

2. 要执行 Fuse 的新安装，请在 JBoss EAP 安装程序上运行 Fuse，如在 JBoss EAP 上安装 指南 中所述。

第 5 章 在 KARAF 独立上升级 FUSE 应用程序

在 Karaf 上升级 Fuse 应用程序：

- 您应该考虑 Apache Camel 更新，如 第 5.1 节 “Camel 迁移注意事项” 所述。
- 您必须更新 Fuse 项目的 Maven 依赖项，以确保您使用正确的 Fuse 版本。

通常，您可以使用 Maven 来构建 Fuse 应用程序。Maven 是 Apache 的免费开源构建工具。Maven 配置在 Fuse 应用程序项目的 pom.xml 文件中定义。在构建 Fuse 项目时，Maven 会搜索外部存储库并下载所需的工件。您可以将 Fuse Bill of Materials (BOM)的依赖关系添加到 pom.xml 文件中，以便 Maven 构建过程会获取正确的 Fuse 支持的工件集合。

以下小节提供有关 Maven 依赖项的信息，以及如何在 Fuse 项目中更新它们。

- 第 5.2 节 “关于 Maven 依赖项”
- 第 5.3 节 “更新 Fuse 项目的 Maven 依赖项”

5.1. CAMEL 迁移注意事项

Red Hat Fuse 使用 Apache Camel 2.23。升级到 Fuse 7.8 时，您应该考虑对 Camel 2.22 和 2.23 的以下更新。

Camel 2.22 更新

- Camel 已从 Spring Boot v1 升级到 v2，因此不再支持 v1。
- 升级到 Spring Framework 5。Camel 应该与 Spring 4.3.x 一起工作，但转发 Spring 5.x 将在以后的版本中是 Spring 5.x 的最低 Spring 版本。
- 升级到 Karaf 4.2。您可以在 Karaf 4.1 上运行 Camel，但在此发行版本中，我们仅官方支持 Karaf 4.2。

- 使用 toD DSL 优化，为可能出现的组件重复使用端点和制作者。例如，基于 HTTP 的组件将重复使用发送到同一主机的动态 URI 的生成者(HTTP 客户端)。

- 现在，具有 read-lock idempotent/idempotent-changed 的 File2 使用者可以被配置为延迟发行任务，以便在处理过程中使用文件扩展窗口，该文件可以在带有共享幂等存储库的主动/主动集群设置中可用，以确保其他节点无法快速将已处理的文件视为可处理的文件（仅当您具有 readLockRemoveOnCommit=true 时）。

- 允许在 request/reply 模式中在 Netty4 producer 上插件自定义请求/回复关联 ID 管理器。author 组件现在默认使用扩展模式来支持大于 140 个字符的 12 月

- REST DSL 生成者现在支持使用 endpointProperties 在 REST 配置中配置。

- Kafka 组件现在支持 HeaderFilterStrategy 插件自定义实现，以控制 Camel 和 Kafka 信息之间的标头映射。

- REST DSL 现在支持客户端请求验证，以验证 Content-Type/Accept 标头是否可以用于 REST 服务。

- Camel 现在有一个 Service Registry SPI，它允许您使用 Camel 实现或 Spring Cloud 将路由注册到服务 registry （如 consul、etcd 或 zookeeper）。

- SEDA 组件现在的默认队列大小为 1000，而不是无限。

- 解决了以下值得注意的问题：

  - 修复了 camel-cxf consumer 的 CXF continuation 超时问题，可能会导致消费者返回带有数据的响应，而不是向调用 SOAP 客户端触发超时。

  - 修复了 camel-cxf consumer 在使用强大的单向操作时，不会发布 UoW。

  - 修复了使用 AdviceWith，并使用 weave onException 等方法无法正常工作。

- 修复了并行处理和流模式中的 Splitter 可能会阻断，而当迭代消息正文（）ator 在第一个调用的 next （）方法调用中抛出异常时。

- 修复了 Kafka 使用者，如果 autoCommitEnable=false，则不会自动提交。

- 修复了文件消费者默认使用 markerFile 作为 read-lock，这应该为 none。

- 修复了使用带有 Kafka 的手动提交以提供当前的记录偏移而不是之前（第一个为 -1 ）。

- 修复了 Java DSL 中的基于内容路由器的问题可能无法解析 when predicates 中的属性占位符。

Camel 2.23 更新

- 升级到 Spring Boot 2.1。

- 现在，可以使用 spring-boot 自动配置来配置其他组件级选项。这些选项包含在 spring-boot 组件元数据 JSON 文件描述符中，以获得工具协助。

- 添加了一个文档部分，其中包含所有组件、数据格式和语言的所有 Spring Boot 自动配置选项。

- 现在，所有 Camel Spring Boot starter JARs 都会在其 JARs 中包含 META-INF/spring-autoconfigure-metadata.properties 文件，以优化 Spring Boot 自动配置。

- Throttler 现在支持基于动态表达式的关联组，以便您可以将消息分组到不同的节流集中。

- Hystrix EIP 现在允许 Camel 的错误处理程序继承，以便在您启用了带有红色错误处理时再次重试整个 Hystrix EIP 块。

- SQL 和 ElSql 用户现在支持路由格式的动态查询参数。请注意，这个功能仅限于使用简单表达式调用 Bean。

- swagger-restdsl maven 插件现在支持从 Swagger 规格文件中生成 DTO 模型类。
- 解决了以下值得注意的问题：
  - 已修复 Aggregator2，使其不会传播用于强制完成所有组的控制标头，因此当路由期间使用另一个聚合器 EIP 时不会再次发生。
  - 修复了错误处理程序中被重新传送的 Tracer 无法正常工作。
  - XML 文档的内置类型转换器可能会输出将错误解析到 stdout，现在已修复为使用日志记录 API 的输出。
  - 修复了使用 charset 选项（如果消息正文基于流传输）的 SFTP 编写文件无法正常工作。
  - 修复了当通过多个路由的路由时，不会重复使用 Zipkin root id，以便将它们分组到一个父范围中。
  - 修复了在主机名包含带有数字的 IP 地址时，使用 HTTP 端点时优化为D。
  - 修复了在通过临时队列进行请求/回复以及使用手动确认模式的 RabbitMQ 的问题。它不会确认临时队列（需要尽可能发出请求/回复）。
  - 修复了在 Allow 标头中为 OPTIONS 请求返回所有允许的 HTTP 动词（如使用 rest-dsl 时）的各种 HTTP 消费者组件。
  - 修复了 FluentProducerTemplate 的 thread-safety 问题。

5.2. 关于 MAVEN 依赖项

Maven Bill of Materials (BOM) 文件的目的是提供一组策展的 Maven 依赖关系版本，它们可以正常工作，从而为您为每个 Maven 工件单独定义版本。

每个容器都有一个专用的 BOM 文件，在其中运行 Fuse。

注意

您可以在此处找到这些 BOM 文件 ：https://github.com/jboss-fuse/redhat-fuse。或者，请参阅 最新的发行注记 以了解有关 BOM 文件更新的信息。

Fuse BOM 提供以下优点：

- 定义 Maven 依赖项的版本，以便在将依赖项添加到 pom.xml 文件时不需要指定版本。
- 定义一组策展的依赖关系，它们被完全测试并支持特定版本的 Fuse。
- 简化 Fuse 的升级.

重要

红帽仅支持由 Fuse BOM 定义的依赖项集合。

5.3. 更新 FUSE 项目的 MAVEN 依赖项

要升级用于 Karaf 的 Fuse 应用，请更新项目的 Maven 依赖项。

流程

1. 打开项目的 pom.xml 文件。
2. 在项目的 pom.xml 文件中添加 dependencyManagement 元素（或者在父 pom.xml 文件中），如下例所示：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<project ...>
  ...
  <properties>
```

```
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <!-- configure the versions you want to use here -->
    <fuse.version>7.8.0.fuse-sb2-780038-redhat-00001</fuse.version>

  </properties>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.jboss.redhat-fuse</groupId>
        <artifactId>fuse-karaf-bom</artifactId>
        <version>${fuse.version}</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>
  ...
</project>
```

3. 保存 pom.xml 文件。

在 pom.xml 文件中将 BOM 指定为依赖项后，可以在 pom.xml 文件中添加 Maven 依赖项，*而无需* 指定工件版本。例如，要为 camel-velocity 组件添加依赖项，您要将以下 XML 片段添加到 pom.xml 文件中的 dependencies 元素中：

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-velocity</artifactId>
  <scope>provided</scope>
</dependency>
```

请注意，此依赖项定义中如何省略 version 元素。

第 6 章 在 KARAF 上升级 FUSE STANDALONE

在这个发行版本中，有许多升级会影响主版本和次发行版本。大多数 OSGi 捆绑包设置版本范围，这些范围排除了下一个主要版本，甚至是次要版本。

警告

不要在 Apache Karaf 补丁机制中使用 Fuse 将 Apache Karaf 容器升级到 Fuse 7.8，必须执行一个新的安装，并将配置和手动复制的其他修改文件。

6.1. 在 KARAF 上升级 FUSE STANDALONE

以下说明指导您在 Apache Karaf 上升级到 Fuse 7.8。

流程

1. 要在 Apache Karaf 上安装 Fuse，请按照在 Apache Karaf 上安装 中的步骤操作。
2. 为 Fuse 7.8 使用新的 Fuse Karaf BOM 重建应用程序（有关 BOM 文件更新的信息，请参阅最新的 发行注记 ）。
3. 安装之前在 Fuse 7.7 中安装的功能
4. 比较用于潜在配置更改的 etc 目录，如日志记录、Web 或 Maven。
5. 比较用于潜在环境更改的 bin 目录，例如 bin/setenv。
6. 将重新构建的应用程序安装到 Fuse 7.8

注意

升级后，您将会在重启容器时看到 Welcome banner 中的新版本和构建号。