



Red Hat AMQ Broker 7.10

AMQ Broker 入门

对于使用 AMQ Broker 7.10

对于使用 AMQ Broker 7.10

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

本指南描述了如何开始使用 AMQ Broker。

目录

使开源包含更多	3
第 1 章 概述	4
1.1. 主要特性	4
1.2. 支持的标准和协议	4
1.3. 支持的配置	4
1.4. 文档惯例	4
第 2 章 了解 AMQ BROKER	6
2.1. 代理实例	6
2.2. 消息持久性	6
2.3. 资源消耗	6
2.4. 监控和管理	7
第 3 章 安装 AMQ BROKER	8
3.1. 下载 AMQ BROKER 归档	8
3.2. 在 LINUX 上提取 AMQ BROKER 归档	8
3.3. 在 WINDOWS 系统中提取 AMQ BROKER 归档	9
3.4. 了解 AMQ BROKER 安装归档内容	9
第 4 章 创建独立代理	10
4.1. 创建代理实例	10
4.2. 启动代理实例	12
4.3. 生成和使用测试信息	12
4.4. 停止 BROKER 实例	14
第 5 章 运行 AMQ BROKER 示例	15
5.1. 将机器设置为运行 AMQ BROKER 示例	15
5.2. AMQ BROKER 示例程序	17
5.3. 运行 AMQ BROKER 示例程序	17
第 6 章 后续步骤	19
附录 A. 使用您的订阅	20
A.1. 访问您的帐户	20
A.2. 激活订阅	20
A.3. 下载发行文件	20
A.4. 为系统注册软件包	20
附录 B. APACHE MAVEN 简介	22
B.1. MAVEN POM 文件	22
B.2. MAVEN 存储库	22
B.3. MAVEN 设置文件	23

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 概述

AMQ Broker 是一个基于 ActiveMQ Artemis 的高性能消息传递的实施。它使用异步日志来实现快速消息持久性，并支持多种语言、协议和平台。

1.1. 主要特性

AMQ Broker 提供以下功能：

- 集群和高可用性选项
- 快速原生IO 持久性
- 支持本地事务
- 在使用 AMQ 核心协议 JMS 和 AMQ OpenWire JMS 客户端时支持 XA 事务
- 使用 Java 编写，用于广泛的平台支持
- 多个管理接口：AMQ 管理控制台、管理 API 和 JMX

1.2. 支持的标准和协议

AMQ Broker 支持以下标准和协议：

- Wire 协议
 - 核心协议
 - AMQP 1.0
 - MQTT
 - OpenWire (由 A-MQ 6 客户端使用)
 - STOMP
- JMS 2.0



注意

AMQP 中的分布式事务(XA)详情在规格的 1.0 版本中没有提供。如果您的环境需要支持分布式事务，建议您使用 AMQ 核心协议 JMS。

1.3. 支持的配置

有关 AMQ Broker 支持的配置的最新信息，请参阅红帽客户门户网站中的["Red Hat AMQ 7 支持的配置"](#)文章。

1.4. 文档惯例

本文档对 **sudo** 命令、文件路径和可替换值使用以下惯例：

sudo 命令

在本文中，**sudo** 用于任何需要 root 特权的命令。使用 **sudo** 时，您应始终谨慎操作，因为任何更改都可能影响整个系统。

有关使用 **sudo** 的更多信息，[请参阅管理 sudo 访问](#)。

关于在此文档中使用文件路径

在这个文档中，所有文件路径都对 Linux、UNIX 和类似操作系统（例如 `/home/...`）有效。如果您使用的是 Microsoft Windows，则应使用等效的 Microsoft Windows 路径（例如，`C:\Users\...`）。

可替换值

本文档有时会使用可替换值，您必须将这些值替换为特定于环境的值。可替换的值为小写，以尖括号 (<>) 括起，样式则使用斜体和 **monospace** 字体。用下划线(_)分隔多个词语。

例如，在以下命令中，将 `<install_dir>` 替换为您自己的目录名称。

```
$ <install_dir>/bin/artemis create mybroker
```

第 2 章 了解 AMQ BROKER

AMQ Broker 使您能够将异构系统松散耦合在一起，同时提供可靠性、事务和许多其他功能。在使用 AMQ Broker 前，您应该了解它提供的功能。

2.1. 代理实例

在 AMQ Broker 中，安装的 AMQ 代理软件充当一个或多个 *代理实例* 的 "home"。这个架构提供了几个优点，例如：

- 您可以从单个 AMQ Broker 安装中创建所需数量的代理实例。
AMQ Broker 安装包含每个代理实例需要运行的必要二进制文件和资源。然后，在代理实例之间共享这些资源。
- 当升级到新版本的 AMQ Broker 时，您只需要更新软件一次，即使您在该主机上运行多个代理实例。

您可以将代理实例视为消息代理。每个代理实例都有自己的目录，其中包含其唯一的配置和运行时数据。此运行时数据由日志和数据文件组成，并与主机上的唯一代理进程相关联。

2.2. 消息持久性

AMQ Broker 会保留消息数据以确保信息永不会丢失，即使代理失败或意外关闭也是如此。AMQ 代理提供两种消息持久性选项：基于日志的持久性和数据库持久性。

基于日志的持久性

默认方法，此选项将消息数据写入文件系统中存储的日志文件。最初，每个日志文件都使用固定大小自动创建，并填充空数据。当客户端执行各种代理操作时，记录将附加到日志中。当其中一个日志文件已满时，代理将移动到下一个日志文件。

基于日志的持久性支持事务操作，包括本地和 XA 事务。

基于日志的持久性需要文件系统的 IO 接口。AMQ Broker 支持以下内容：

Linux 异步 IO(AIO)

AIO 通常提供最佳的性能。确保您的网络文件系统在 [Red Hat AMQ 7 支持的配置](#) 中被列为支持。

Java NIO

Java NIO 提供了良好的性能，它可以在任何具有 Java 6 或更高版本运行时的平台中运行。

数据库持久性

此选项通过使用 Java 数据库连接(JDBC)将消息和绑定数据存储存储在数据库中。如果您的环境中已有可靠和高性能的数据库平台，或者公司政策强制使用数据库，则此选项是一个不错的选择。

代理 JDBC 持久存储使用标准 JDBC 驱动程序来创建 JDBC 连接，它将消息和绑定数据存储存储在数据库表中。数据库表中的数据采用与基于日志的持久性相同的编码方式进行编码。这意味着，如果直接使用 SQL 访问数据库中存储的消息，则不会被人读取。

要使用数据库持久性，您必须使用受支持的数据库平台。要查看当前支持的数据库平台，请参阅 [Red Hat AMQ 7 支持的配置](#)。

2.3. 资源消耗

AMQ Broker 提供了多个选项来限制代理的内存和资源消耗。

资源限值

您可以为每个用户设置连接和队列限制。这可防止用户消耗太多代理的资源，并导致其他用户的性能下降。

消息分页

消息分页可让 AMQ Broker 支持包含数百万消息的大型队列，同时使用有限内存运行。代理收到大量超过其内存容量的消息时，它开始将消息分页到磁盘。此分页过程透明；根据需要，代理页面消息进入内存和内存不足。

消息分页基于地址。当地址内存中所有消息的大小超过最大大小时，该地址的每条附加消息都将分页到该地址的页面文件中。

大消息

使用 AMQ Broker，您可以发送和接收大量信息，即使使用有限内存资源运行时也是如此。为避免在内存中存储大量消息的开销，您可以配置 AMQ Broker 以将这些大信息存储在文件系统或数据库表中。

2.4. 监控和管理

AMQ Broker 提供了几个用来监控和管理代理的工具。

AMQ 管理控制台

AMQ 管理控制台是可通过 Web 浏览器访问的 Web 界面。您可以使用它来监控网络健康、查看代理拓扑，以及创建和删除代理资源。

CLI

AMQ Broker 提供 **artemis** CLI，可用于管理代理。使用 CLI，您可以创建、启动和停止代理实例。CLI 也提供多个命令来管理消息日志。

管理 API

AMQ Broker 提供了广泛的管理 API。您可以使用它来修改代理的配置、创建新资源、检查这些资源并与它们交互。客户端也可以使用管理 API 管理代理并订阅管理通知。

AMQ Broker 提供以下使用管理 API 的方法：

- Java 管理扩展(JMX)- JMX 是用于管理 Java 应用程序的标准技术。代理的管理操作通过 AMQ MBeans 接口公开。
- JMS API - 使用标准 JMS 消息将管理操作发送到特殊的管理 JMS 队列。

日志

每个代理实例记录错误消息、警告和其他与代理相关的信息和活动。您可以配置日志记录级别、日志文件的位置和日志格式。然后，您可以使用生成的日志文件监控代理并诊断错误条件。

第 3 章 安装 AMQ BROKER

AMQ Broker 作为独立于平台的归档文件进行分发。要在您的系统上安装 AMQ Broker，您必须下载存档并提取内容。您还应了解存档中包含的目录。

先决条件

- 安装 AMQ Broker 的主机必须满足 AMQ Broker 支持的配置。
如需更多信息，请参阅 [支持的 Red Hat AMQ 7 配置](#)。

3.1. 下载 AMQ BROKER 归档

AMQ Broker 作为独立于平台的归档文件进行分发。您可从红帽客户门户下载。

先决条件

- 您必须拥有红帽订阅。
如需更多信息，请参阅 [使用您的订阅](#)。

流程

1. 在 Web 浏览器中，导航到 <https://access.redhat.com/downloads/> 并登录。
此时会显示产品下载页。
2. 在 **JBoss 集成和自动化** 部分中，点击 **Red Hat AMQ Broker** 链接。
此时会显示 **Software Downloads** 页面。
3. 从 **Version** 下拉菜单中选择所需的 AMQ Broker 版本。
4. 在 **Releases** 选项卡中，点您要下载的特定 AMQ Broker 文件的 **Download** 链接。

3.2. 在 LINUX 上提取 AMQ BROKER 归档

如果您要在 Red Hat Enterprise Linux 上安装 AMQ Broker，请为 AMQ Broker 创建新用户帐户，然后从安装存档中提取内容。

流程

1. 创建一个名为 **amq-broker** 的新用户，并提供密码。

```
$ sudo useradd amq-broker  
$ sudo passwd amq-broker
```

2. 创建目录 **/opt/redhat/amq-broker**，并使其成为新 **amq-broker** 用户和组。

```
$ sudo mkdir /opt/redhat  
$ sudo mkdir /opt/redhat/amq-broker  
$ sudo chown -R amq-broker:amq-broker /opt/redhat/amq-broker
```

3. 将存档的所有者更改为新用户。

```
$ sudo chown amq-broker:amq-broker amq-broker-7.x.x-bin.zip
```

4. 将安装存档移至您刚才创建的目录。

```
$ sudo mv amq-broker-7.x.x-bin.zip /opt/redhat/amq-broker
```

5. 以新的 **amq-broker** 用户身份，使用 **unzip** 命令提取内容。

```
$ su - amq-broker
$ cd /opt/redhat/amq-broker
$ unzip <archive_name>.zip
$ exit
```

创建名称与 **amq-broker-7.10** 类似的目录。在文档中，这个位置被称为 **<install_dir>**。

3.3. 在 WINDOWS 系统中提取 AMQ BROKER 归档

如果要在 Windows 系统上安装 AMQ Broker，请为 AMQ Broker 创建新目录文件夹，然后在其中提取内容。

流程

1. 使用 Windows Explorer 在所需的驱动器符上创建目录文件夹 **\redhat\amq-broker**。
例如：**C:\redhat\amq-broker**
2. 使用 Windows Explorer 将安装存档移至您刚才创建的目录。
3. 在 **\redhat\amq-broker** 目录中，右键单击安装存档 zip 文件并选择 **Extract All**。
创建名称与 **amq-broker-7.10** 类似的目录。在文档中，这个位置被称为 **<install_dir>**。

3.4. 了解 AMQ BROKER 安装归档内容

通过提取存档创建的目录是 AMQ Broker 安装的顶级目录。该目录被称为 **<install_dir>**，包括以下内容：

此目录...	包含...
<install_dir>/web/api	API 文档.
<install_dir>/bin	运行 AMQ Broker 所需的二进制文件和脚本.
<install_dir>/etc	配置文件.
<install_dir>/examples	JMS 和 Java EE 示例.
<install_dir>/lib	运行 AMQ Broker 所需的 jar 和库.
<install_dir>/schema	用于验证 AMQ 代理配置的 XML 模式.
<install_dir>/web	AMQ Broker 运行时加载的 web 上下文.

第 4 章 创建独立代理

您可以通过在本地机器上创建独立代理实例，启动它，并生成和使用一些测试信息来快速使用 AMQ Broker。

先决条件

- 必须安装 AMQ Broker。
更多信息请参阅 [第 3 章 安装 AMQ Broker](#)。

4.1. 创建代理实例

代理实例是一个目录，其中包含代理的配置和运行时数据。要创建新的代理实例，您首先为 broker 实例创建一个目录，然后使用 **artemis create** 命令创建代理实例。

此流程演示了如何在本地机器中创建简单独立代理。代理使用基本的默认配置，并接受使用任何受支持的消息传递协议来自客户端的连接。

流程

1. 为 broker 实例创建一个目录。

如果使用...	这些以下操作
Red Hat Enterprise Linux	<ol style="list-style-type: none"> 1. 创建一个新目录，以用作 broker 实例的位置。 <pre>\$ sudo mkdir /var/opt/amq-broker</pre> 2. 分配安装期间创建的用户。 <pre>\$ sudo chown -R amq-broker:amq-broker /var/opt/amq-broker</pre>
Windows	使用 Windows Explorer 创建一个新文件夹，以用作代理实例的位置。

2. 使用 **artemis create** 命令创建代理。

如果使用...	这些以下操作
---------	--------

如果使用...	这些以下操作
Red Hat Enterprise Linux	<ol style="list-style-type: none"> 1. 切换到安装期间创建的用户帐户。 <pre>\$ su - amq-broker</pre> 2. 更改到您刚才为代理实例创建的目录。 <pre>\$ cd /var/opt/amq-broker</pre> 3. 从 broker 实例的目录中，创建 broker 实例。 <pre>\$ <install_dir>/bin/artemis create mybroker</pre>
Windows	<ol style="list-style-type: none"> 1. 从您刚才为代理实例创建的目录打开命令提示符。 2. 从 broker 实例的目录中，创建 broker 实例。 <pre>> <install_dir>\bin\artemis.cmd create mybroker</pre>

3. 按照 **artemis create** 提示来配置代理实例。

例 4.1. 使用 **artemis create** 配置代理实例

```
$ /opt/redhat/amq-broker/bin/artemis create mybroker

Creating ActiveMQ Artemis instance at: /var/opt/amq-broker/mybroker

--user: is mandatory with this configuration:
Please provide the default username:
admin

--password: is mandatory with this configuration:
Please provide the default password:

--role: is mandatory with this configuration:
Please provide the default role:
amq

--allow-anonymous | --require-login: is mandatory with this configuration:
Allow anonymous access? (Y/N):
Y

Auto tuning journal ...
done! Your system can make 19.23 writes per millisecond, your journal-buffer-timeout will
be 52000

You can now start the broker by executing:

"/var/opt/amq-broker/mybroker/bin/artemis" run
```


启动代理后，您应该验证其是否在正确运行。这涉及生成一些测试信息，将它们发送到代理，然后使用它们。

流程

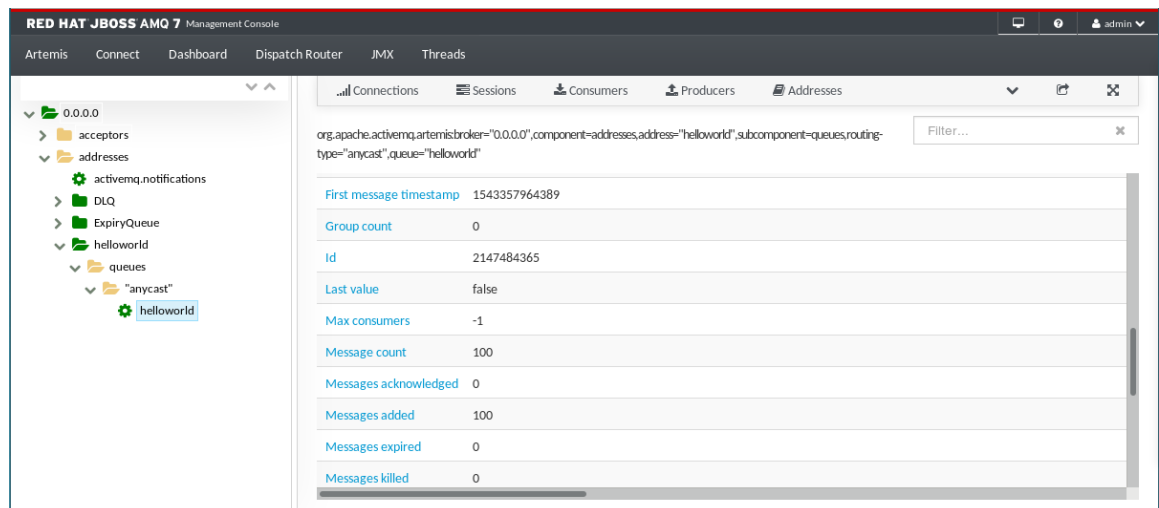
1. 使用 **artemis producer** 命令生成几个测试消息，并将它们发送到代理。
此命令会向 **helloworld** 地址发送 100 条消息，该地址会在代理上自动创建。生产者使用默认端口 61616 连接到代理，该端口接受所有支持的消息传递协议。

```
$ /opt/redhat/amq-broker/amq-broker-7.2.0/bin/artemis producer --destination helloworld --
message-count 100 --url tcp://localhost:61616
Producer ActiveMQQueue[helloworld], thread=0 Started to calculate elapsed time ...
```

```
Producer ActiveMQQueue[helloworld], thread=0 Produced: 100 messages
Producer ActiveMQQueue[helloworld], thread=0 Elapsed time in second : 1 s
Producer ActiveMQQueue[helloworld], thread=0 Elapsed time in milli second : 1289 milli
seconds
```

2. 使用 Web 控制台查看存储在代理中的消息。
 - a. 在 Web 浏览器中，导航到 <http://localhost:8161>
 - b. 使用您在创建代理实例时创建的默认用户名和密码登录控制台。
此时会显示 **Attributes** 选项卡。
 - c. 在 **Attributes** 选项卡上，导航到 **addresses** → **helloworld** → **"anycast"** → **helloworld**。
在上一步中，您发送了消息到 **helloworld** 地址。这会使用队列（也称为 **helloworld**）创建新的任播 **helloworld** 地址。**Message count** 属性显示发送到 **helloworld** 的所有 100 个消息目前都存储在此队列中。

图 4.1. 消息计数



3. 使用 **artemis consumer** 命令，使用存储在代理中的 50 个消息。
这个命令会使用您之前发送到代理的 50 个信息。

```
$ /opt/redhat/amq-broker/amq-broker-7.2.0/bin/artemis consumer --destination helloworld --
message-count 50 --url tcp://localhost:61616
```

```
Consumer:: filter = null
```

```
Consumer ActiveMQQueue[helloworld], thread=0 wait until 50 messages are consumed
Consumer ActiveMQQueue[helloworld], thread=0 Consumed: 50 messages
Consumer ActiveMQQueue[helloworld], thread=0 Consumer thread finished
```

4. 在 Web 控制台中，验证 **Message 计数** 现在是否为 50。
50 个消息被使用，保留 50 个消息存储在 **helloworld** 队列中。
5. 停止代理，并验证剩余的 50 个消息仍然存储在 **helloworld** 队列中。
 - a. 在代理运行的终端中，按 **Ctrl+C** 以停止代理。
 - b. 重新启动代理。

```
$ /var/opt/amq-broker/mybroker/bin/artemis run
```

- c. 在 Web 控制台中，导航到 **helloworld** 队列，再验证队列中仍然存储了 50 个消息。
6. 使用剩余的 50 条消息。

```
$ /opt/redhat/amq-broker/amq-broker-7.2.0/bin/artemis consumer --destination helloworld --
message-count 50 --url tcp://localhost:61616
```

```
Consumer:: filter = null
Consumer ActiveMQQueue[helloworld], thread=0 wait until 50 messages are consumed
Consumer ActiveMQQueue[helloworld], thread=0 Consumed: 50 messages
Consumer ActiveMQQueue[helloworld], thread=0 Consumer thread finished
```

7. 在 Web 控制台中，验证 **Message count** 是否为 0。
helloworld 队列中存储的所有消息都被使用，队列现在为空。

4.4. 停止 BROKER 实例

创建独立代理并生成和使用测试消息后，您可以停止代理实例。

此流程手动停止代理，代理强制关闭所有客户端连接。在生产环境中，您应该将代理配置为正常停止，以便正确关闭客户端连接。

流程

- 使用 **artemis stop** 命令停止代理实例：

```
$ /var/opt/amq-broker/mybroker/bin/artemis stop
2018-12-03 14:37:30,630 INFO [org.apache.activemq.artemis.core.server] AMQ221002:
Apache ActiveMQ Artemis Message Broker version 2.6.1.amq-720004-redhat-1 [b6c244ef-
f1cb-11e8-a2d7-0800271b03bd] stopped, uptime 35 minutes
Server stopped!
```

第 5 章 运行 AMQ BROKER 示例

AMQ Broker 随附许多示例程序，它们演示了该产品的基本和高级功能。您可以运行这些示例来熟悉 AMQ Broker 的功能。

要运行 AMQ Broker 示例，您必须首先通过安装和配置 Apache Maven 和 AMQ Maven 存储库来设置机器。然后，您可以使用 Maven 运行 AMQ Broker 示例程序。

5.1. 将机器设置为运行 AMQ BROKER 示例

在运行附带的 AMQ Broker 示例程序前，您必须先下载并安装 Maven 和 AMQ Maven 存储库，并配置 Maven 设置文件。

5.1.1. 下载并安装 Maven

需要 Maven 运行 AMQ Broker 示例。

流程

1. 进入 [Apache Maven 下载页面](#)，再下载您的操作系统的最新发行版。
2. 为您的操作系统安装 Maven。
如需更多信息，请参阅 [安装 Apache Maven](#)。

其他资源

- 有关 Maven 的更多信息，请参阅 [Apache Maven 简介](#)。

5.1.2. 下载并安装 AMQ Maven 存储库

在您的机器上安装 Maven 后，您将下载并安装 AMQ Maven 存储库。红帽客户门户网站中提供这个软件仓库。

1. 在 Web 浏览器中，导航到 <https://access.redhat.com/downloads/> 并登录。
此时会显示产品下载页。
2. 在 **Integration and Automation** 部分中，点 **Red Hat AMQ Broker** 链接。
此时会显示 **Software Downloads** 页面。
3. 从 **Version** 下拉菜单中选择所需的 AMQ Broker 版本。
4. 在 **Releases** 选项卡上，点 AMQ Broker Maven Repository 的 **Download** 链接。
AMQ Maven 存储库文件作为 zip 文件下载。
5. 在您的计算机上，将 AMQ 存储库文件解压缩到您选择的目录中。
在您的计算机上创建新的目录，其中包含名为 **maven-repository/** 的子目录中的 Maven 存储库。

5.1.3. 配置 Maven 设置文件

下载并安装 AMQ Maven 存储库后，您必须将存储库添加到 Maven 设置文件中。

流程

1. 打开 Maven **settings.xml** 文件。
settings.xml 文件通常位于 `${user.home}/.m2/` 目录中。
 - 对于 Linux, 这是 `~/.m2/`。
 - 对于 Windows, 这是 `\Documents and Settings\.m2\` 或 `\Users\.m2\`

如果您未在 `${user.home}/.m2/` 中找到 **settings.xml** 文件, 则 Maven 安装的 `conf/` 目录中有一个默认版本。将默认 **settings.xml** 文件复制到 `${user.home}/.m2/` 目录中。

2. 在 `<profiles>` 元素中, 为 AMQ Maven 存储库添加配置集。

```

<!-- Configure the JBoss AMQ Maven repository -->
<profile>
  <id>jboss-amq-maven-repository</id>
  <repositories>
    <repository>
      <id>jboss-amq-maven-repository</id>
      <url>file://<JBoss-AMQ-repository-path></url> 1
      <releases>
        <enabled>>true</enabled>
      </releases>
      <snapshots>
        <enabled>>false</enabled>
      </snapshots>
    </repository>
  </repositories>
  <pluginRepositories>
    <pluginRepository>
      <id>jboss-amq-maven-repository</id>
      <url>file://<JBoss-AMQ-repository-path></url> 2
      <releases>
        <enabled>>true</enabled>
      </releases>
      <snapshots>
        <enabled>>false</enabled>
      </snapshots>
    </pluginRepository>
  </pluginRepositories>
</profile>

```

- 1 2 将 `<JBoss-AMQ-repository-path>` 替换为您安装的 Maven 存储库的位置。通常, 此位置以 `/maven-repository` 结尾。例如 :

```

<url>file:///path/to/repo/amq-broker-7.2.0-maven-repository/maven-repository</url>

```

3. 在 `<activeProfiles>` 元素中, 将 AMQ Maven 存储库设置为活跃 :

```

<activeProfiles>
  <activeProfile>jboss-amq-maven-repository</activeProfile>
  ...
</activeProfiles>

```

4. 如果您从 Maven 安装中复制了默认的 **settings.xml**，取消注释 **<active-profiles>** 部分（如果默认被注释掉）。
5. 保存并关闭 **settings.xml**。
6. 删除缓存的 **`\${user.home}/.m2/repository/`** 目录。
如果您的 Maven 存储库包含过时的工件，则构建或部署项目时可能会遇到以下 Maven 错误消息之一：
 - 缺少工件 **<artifact-name>**
 - **[ERROR] Failed to execute goal on project <project-name>; Could not resolve dependencies for <project-name>**

5.2. AMQ BROKER 示例程序

AMQ Broker 附带 90 多个示例程序，演示了如何使用 AMQ Broker 功能和 supported 的消息传递协议。

示例程序位于 **<install_dir>/examples**，并包括以下程序：

- 功能
特定于代理的功能，例如：
 - Clustered - 显示负载均衡和分布功能的示例
 - HA - 显示故障转移和重新连接功能的示例
 - Perf - 允许您在服务器上运行几个性能测试的示例
 - Standard - 演示各种代理功能的示例
 - Sub-modules - 集成外部模块示例
- 协议
每个支持的消息传递协议示例：
 - AMQP
 - MQTT
 - OpenWire
 - STOMP

其他资源

- 有关每个示例程序的描述，请参阅 Apache Artemis 文档中的 [示例](#)。

5.3. 运行 AMQ BROKER 示例程序

AMQ Broker 随附许多示例程序，它们演示了该产品的基本和高级功能。您可以使用 Maven 运行这些程序。

先决条件

- 必须设置您的机器才能运行 AMQ Broker 示例。

更多信息请参阅 [第 5.1 节 “将机器设置为运行 AMQ Broker 示例”](#)。

流程

1. 导航到您要运行的示例的目录。
示例程序位于 `<install_dir>/examples` 中。例如：

```
$ cd <install_dir>/examples/features/standard/queue
```

2. 使用 `mvn clean verify` 命令来运行示例程序。
Maven 启动代理并运行示例程序。第一次运行示例程序时，Maven 会下载任何缺少的依赖项，这可能需要一些时间才能运行。

在这种情况下，队列示例程序会运行，它创建一个制作者，发送测试消息，然后创建一个接收消息的消费者：

```
$ mvn clean verify
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.apache.activemq.examples.broker:queue >-----
[INFO] Building ActiveMQ Artemis JMS Queue Example 2.6.1.amq-720004-redhat-1
[INFO] -----[ jar ]-----
...
server-out:2018-12-05 16:37:57,023 INFO [org.apache.activemq.artemis.core.server]
AMQ221001: Apache ActiveMQ Artemis Message Broker version 2.6.1.amq-720004-redhat-1
[0.0.0.0, nodeID=06f529d3-f8d6-11e8-9bea-0800271b03bd]
[INFO] Server started
[INFO]
[INFO] --- artemis-maven-plugin:2.6.1.amq-720004-redhat-1:runClient (runClient) @ queue --
-
Sent message: This is a text message
Received message: This is a text message
[INFO]
[INFO] --- artemis-maven-plugin:2.6.1.amq-720004-redhat-1:cli (stop) @ queue ---
server-out:2018-12-05 16:37:59,519 INFO [org.apache.activemq.artemis.core.server]
AMQ221002: Apache ActiveMQ Artemis Message Broker version 2.6.1.amq-720004-redhat-1
[06f529d3-f8d6-11e8-9bea-0800271b03bd] stopped, uptime 3.734 seconds
server-out:Server stopped!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 48.681 s
[INFO] Finished at: 2018-12-05T16:37:59-05:00
[INFO] -----
```



注意

一些示例程序使用 UDP 集群，默认情况下可能无法在您的环境中工作。要成功运行这些示例，请将定向到 224.0.0.0 的流量重定向到回环接口：

```
$ sudo route add -net 224.0.0.0 netmask 240.0.0.0 dev lo
```

第 6 章 后续步骤

安装 AMQ Broker 并使用默认配置设置创建独立代理后，您可以将其配置为满足消息传递要求，将消息传递客户端应用程序连接到其中，并监控和管理它。还提供其他资源来帮助您完成这些目标。

配置代理

使用 [配置 AMQ Broker](#) 配置代理以满足您的要求。您可以配置：

- 接受客户端连接的代理
- 地址空间（包括点对点和发布订阅消息传递）
- 消息持久性
- 代理资源消耗（包括资源限值、消息分页和大型消息支持）
- 重复消息检测
- 日志记录

保护代理

使用 [配置 AMQ Broker](#) 来实现以下任一方法来保护代理：

- 客户机/匿名访问控制
- 基本用户和密码访问控制
- 基于证书的访问控制
- LDAP 集成
- Kerberos 集成

设置集群和高可用性

使用 [配置 AMQ Broker](#) 添加额外代理来组成代理集群并提高消息传递吞吐量。您还可以配置高可用性以提高消息传递可靠性。

创建消息传递客户端应用程序

使用 [AMQ 客户端概述](#) 了解 AMQ 客户端及其如何帮助您创建连接到代理以及生成和使用消息的消息传递客户端应用。

监控和管理代理

使用 [管理 AMQ Broker](#) 来监控和管理代理（或代理）运行后。

附录 A. 使用您的订阅

AMQ 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。

A.1. 访问您的帐户

流程

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。

A.2. 激活订阅

流程

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。

A.3. 下载发行文件

要访问 .zip、.tar.gz 和其他发布文件，请使用客户门户查找要下载的相关文件。如果您使用 RPM 软件包或 Red Hat Maven 存储库，则不需要这一步。

流程

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. 查找 **INTEGRATION** 目录中的红帽 AMQ 条目。
3. 选择所需的 AMQ 产品。此时会打开 **Software Downloads** 页面。
4. 单击组件的 **Download** 链接。

A.4. 为系统注册软件包

要在 Red Hat Enterprise Linux 上安装此产品的 RPM 软件包，必须注册您的系统。如果您使用下载的发行文件，则不需要这一步。

流程

1. 转至 access.redhat.com。
2. 进入 **Registration Assistant**。
3. 选择您的操作系统版本，再继续到下一页。
4. 使用您的系统终端中列出的命令完成注册。

有关注册您的系统的更多信息，请参阅以下资源之一：

- [Red Hat Enterprise Linux 7 - 注册系统并管理订阅](#)
- [Red Hat Enterprise Linux 8 - 注册系统并管理订阅](#)

附录 B. APACHE MAVEN 简介

Apache Maven 是 Java 应用程序开发中使用的分布式构建自动化工具，用于创建、管理和构建软件项目。您可以使用它来运行 AMQ Broker 安装中包含的 AMQ Broker 示例程序。

运行 AMQ Broker 示例程序需要您与几个 Maven 组件交互：

项目对象模型(POM)文件

存储关于项目构建方式的信息。

软件仓库

包含构建工件和依赖项。

Maven 设置文件

存储用户特定配置信息。

B.1. MAVEN POM 文件

Maven 使用名为 Project Object Model(POM)文件的标准配置文件来定义项目并管理构建流程。它们确保正确而一致地构建项目。POM 文件是 XML 文件(**pom.xml**)。

Maven 优先选择"交付而非配置"。因此，POM 文件需要最少的配置和默认的所有其他值。POM 文件可以为 Maven 项目定义以下信息：

- 源、测试和目标目录的位置
- 项目依赖项
- 插件存储库
- 项目可以执行的目标
- 有关项目的其他详细信息，如版本、描述、开发人员、邮件列表和许可证等。

例 B.1. pom.xml 文件示例

这个基本 **pom.xml** 文件演示了 POM 文件的最低要求：

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.jboss.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1</version>
</project>
```

其它资源

- **pom.xml** 文件的 schema 位于 http://maven.apache.org/maven-v4_0_0.xsd。
- 有关 POM 文件的更多信息，请参阅 [Apache Maven Project POM 参考](#)。

B.2. MAVEN 存储库

Maven 存储库存储 Java 库、插件和其他构建构件。存储库可以是本地或远程存储库。

默认公共存储库是 [Maven 2 Central Repository](#)，但存储库可以是私有的和内部的，以便在开发团队之间共享通用工件。

也可从第三方获取存储库。AMQ 包括一个 Maven 存储库，其中包含 AMQ 7.10 Java 软件包和依赖项的经过测试和支持版本。

其他资源

- 如需有关 Maven 存储库的更多信息，请参阅 [存储库简介](#)。

B.3. MAVEN 设置文件

Maven **settings.xml** 文件包含 Maven 的特定用户配置信息。它包含的信息不得通过 **pom.xml** 文件分发，如开发人员身份、代理信息、本地存储库位置，以及特定于用户的其他设置。

settings.xml 文件位于两个位置：

- 在 Maven 安装中：
settings.xml 文件位于 `<maven-install-dir>/conf/` 目录中。这些设置称为 **全局** 设置。默认的 Maven 设置文件是一个模板，您可以复制该模板，并用作用户设置文件的起点。
- 在用户安装中：
settings.xml 文件位于 `${user.home}/.m2/` 目录中。如果 Maven 和用户 **settings.xml** 文件都存在，则内容将合并。如果存在重叠，用户的 **settings.xml** 文件优先。

例 B.2. Maven 设置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <profiles>
    <!-- Configure the JBoss AMQ Maven repository -->
    <profile>
      <id>jboss-amq-maven-repository</id>
      <repositories>
        <repository>
          <id>jboss-amq</id>
          <url>file:///path/to/repo/</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>jboss-amq-maven-plugin-repository</id>
          <url>file:///path/to/repo/</url>
          <releases>
```

```
<enabled>true</enabled>
</releases>
<snapshots>
  <enabled>>false</enabled>
</snapshots>
</pluginRepository>
</pluginRepositories>
</profile>
</profiles>
<activeProfiles>
  <!-- Optionally, make the repository active by default -->
  <activeProfile>jboss-amq-maven-repository</activeProfile>
</activeProfiles>
</settings>
```

其他资源

- **settings.xml** 文件的 schema 位于 <http://maven.apache.org/xsd/settings-1.0.0.xsd>。
- 有关 Maven 设置文件的更多信息，请参阅 [Settings 参考](#)。

更新于 2024-06-11