Red Hat JBoss Core Services 2.4.57

Apache HTTP 服务器连接器和负载平衡指南

用于 Red Hat JBoss Core Services 2.4.57

Last Updated: 2024-03-20

Red Hat JBoss Core Services 2.4.57 Apache HTTP 服务器连接器和负载平衡指南

用于 Red Hat JBoss Core Services 2.4.57

法律通告

摘要

安装和配置使用 mod_jk 和 mod_proxy_cluster HTTP 连接器以及 Red Hat JBoss Core Services 提供的其他模块的负载平衡解决方案。

目录

提供有关 RED HAT JBOSS CORE SERVICES 文档的反馈

要报告错误或改进文档，请登录到 Red Hat JIRA 帐户并提交问题。如果您没有 Red Hat Jira 帐户，则会提示您创建一个帐户。

流程

1. 单击以下链接 **以创建 ticket**。
2. 在 **Summary** 中输入问题的简短描述。
3. 在 **Description** 中提供问题或功能增强的详细描述。包括一个指向文档中问题的 URL。
4. 点 **Submit** 创建问题，并将问题路由到适当的文档团队。

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 HTTP 连接器

红帽 JBoss 核心服务(JBCS)包括两种不同的 HTTP 连接器，Apache HTTP 服务器可用于对一组后端 servlet 容器进行负载平衡 HTTP 请求：

- Apache Tomcat 连接器(**mod_jk**) 支持 HTTP 请求对一组 servlet 容器的负载平衡，同时维护粘性会话并通过 Apache JServ 协议 (AJP) 进行通信。
- JBoss HTTP 连接器 (**mod_proxy_cluster**) 是比 **mod_jk** 更高级的负载均衡器。**mod_proxy_cluster** 连接器提供 **mod_jk** 的所有功能，以及实时负载平衡计算、应用生命周期控制、自动代理发现和多个协议支持等功能。

JBCS 和 Red Hat Enterprise Linux (RHEL)提供 Apache HTTP 服务器的独立分发。您可以使用 Apache HTTP 服务器的 JBCS 发行版，方法是使用 **mod_jk**、mod_proxy_**cluster 或 mod_proxy** 连接器作为代理连接到后端应用服务器。

请考虑以下指南：

- 从 RHEL 7 开始，Apache HTTP 服务器的 JBCS 和 RHEL 发行版提供相同的 **mod_proxy** 模块。
- 在 RHEL 版本 7 和 8 中，只有 Apache HTTP 服务器的 JBCS 发行版提供 **mod_jk** 和 **mod_proxy_cluster** 连接器。
- 从 RHEL 9 开始，Apache HTTP 服务器的 JBCS 和 RHEL 发行版还提供与 **mod_jk** 连接器和 **mod_proxy_cluster** 连接器相同的副本。
- 从存档文件或从 RPM 软件包安装 Apache HTTP 服务器的 JBCS 发行版，使用 **groupinstall** 选项还会自动安装 **mod_jk** 和 **mod_proxy_cluster** 连接器。
- 安装 Apache HTTP 服务器的 RHEL 9 发行版不会自动安装 **mod_jk** 和 **mod_proxy_cluster** 连接器。在这种情况下，您可以使用 RHEL Application Streams 手动安装适当的 **mod_jk** 或 **mod_proxy_cluster** 软件包。如需更多信息，请参阅 **Mod_jk 安装和升级** 。

Apache HTTP 服务器连接器和负载平衡指南描述了如何安装和配置 JBCS 提供的 **mod_jk** 和 **mod_proxy_cluster** 连接器。本指南还包括使用 mod_proxy_cluster 进行基本负载平衡的工作示例。

重要

本指南中显示的大多数文件和目录路径都是 Red Hat Enterprise Linux 上 JBCS 的归档安装。对于其他平台，请使用您相应安装的正确路径，如 Red Hat JBoss Core Services Apache HTTP Server 安装指南中的 指定。

其他资源

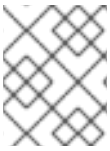
- 使用 Apache Tomcat 连接器(**mod_jk**)的负载平衡
- 使用 JBoss HTTP 连接器(**mod_proxy_cluster**)进行负载平衡

第 2 章 使用 APACHE TOMCAT 连接器 (MOD_JK) 的负载平衡

Apache Tomcat Connector, **mod_jk** 是一个插件，它允许 Apache HTTP 服务器将 Web 请求转发到后端 servlet 容器。**mod_jk** 模块还允许 Apache HTTP 服务器将请求负载平衡到一组 servlet 容器，同时维护粘性会话。

2.1. MOD_JK 安装

红帽 JBoss 核心服务(JBCS)和 Red Hat Enterprise Linux (RHEL)提供单独的 Apache HTTP 服务器分发。您安装的 Apache HTTP 服务器分发决定了 **mod_jk** 连接器的安装是自动的，还是需要手动步骤。根据您安装的 Apache HTTP 服务器的分发，**mod_jk** 模块和配置文件的安装路径也会不同。



注意

JBCS Apache HTTP 服务器支持在所有支持的操作系统中使用 **mod_jk**。RHEL Apache HTTP 服务器只支持在 RHEL 9 上使用 **mod_jk**。

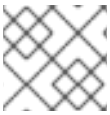
2.1.1. 使用 JBCS Apache HTTP 服务器安装 mod_jk

JBCS 安装的 Apache HTTP 服务器部分会自动安装 **mod_jk** 模块。

您可以按照 Red Hat JBoss Core Services Apache HTTP Server 安装指南中的步骤为您的操作系统安装 JBCS Apache HTTP 服务器。如需更多信息，**请参阅附加资源** 链接。

在使用 JBCS Apache HTTP 服务器时，请考虑以下有关 **mod_jk** 安装的准则：

- **mod_jk.so** 模块安装在 ***JBCS_HOME*/httpd/modules** 目录中。
- **mod_jk.conf.sample**、**worker.properties.sample**、**urworkermap.properties.sample** 配置文件位于 ***JBCS_HOME*/httpd/conf.d** 目录中。
- **mod_jk.conf.sample** 文件包含 **mod_jk** 模块的 **LoadModule** 指令。



注意

JBCS_HOME 代表 JBCS 安装的顶级目录，即 **/opt/jbcs-httpd24-2.4**。

其他资源

- 从归档文件在 RHEL 上安装 JBCS Apache HTTP 服务器
- 从 RPM 软件包在 RHEL 7 或 RHEL 8 上安装 JBCS Apache HTTP 服务器
- 在 Windows Server 上安装 JBCS Apache HTTP 服务器

2.1.2. 使用 RHEL Application Streams 安装 mod_jk

如果您使用 Application Streams 从 RPM 软件包安装 Apache HTTP 服务器的 RHEL 9 发行版，RHEL 不会自动安装 **mod_jk** 软件包。在这种情况下，如果要使用 **mod_jk** 连接器，您必须手动安装 **mod_jk** 软件包。

先决条件

- 已使用 Application Streams 在 RHEL 9 上安装了 Apache HTTP 服务器。

流程

- 以 root 用户身份输入以下命令：

```
# dnf install mod_jk
```

验证

- 要检查 **mod_jk** 软件包是否已成功安装，请输入以下命令：

```
# rpm -q mod_jk
```

  前面的命令输出已安装软件包的全名，其中包括版本和平台信息。

在使用 RHEL Application Streams 时，请考虑以下有关 **mod_jk** 安装的准则：

- **mod_jk.so** 模块安装在 **/usr/lib64/httpd/modules** 目录中。
- **mod_jk.conf.sample**、**worker.properties.sample**、**urworkermap.properties.sample** 配置文件位于 **/etc/httpd/conf.d** 目录中。
- **mod_jk.conf.sample** 文件包含 **mod_jk** 模块的 **LoadModule** 指令。

其他资源

- 应用程序流
- 使用 DNF 工具管理软件

2.2. 使用 MOD_JK 时 APACHE HTTP 服务器负载均衡配置

您可以将 Apache HTTP 服务器配置为使用 **mod_jk** 连接器将请求负载平衡到一组 servlet 容器。此设置包括后端 worker 节点的配置。

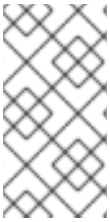
根据您是否通过 Red Hat JBoss Core Services (JBCS)或使用 Red Hat Enterprise Linux (RHEL) Application Streams 安装 **mod_jk**，请考虑以下指南：

- JBCS 在 ***JBCS_HOME*/httpd/conf.d/ 目录中提供 mod_ jk** 的示例配置文件。
- RHEL 在 **/etc/httpd/conf.d/** 目录中提供了 **mod_jk** 的示例配置文件。

mod_jk 的示例配置文件名为 **mod_jk.conf.sample**、**worker.properties.sample**、**uriworkermap.properties.sample**。要使用这些示例而不是创建自己的配置文件，您可以删除 **.sample** 扩展，并根据需要修改文件内容。

注意

您还可以使用红帽客户门户网站中的 Load Balancer 配置工具，快速为 **mod_jk** 和 Tomcat worker 节点生成最佳配置模板。当您将 Load Balancer Configuration 工具用于 Apache HTTP Server 2.4.57 时，请确保选择 **2.4.x** 作为 Apache 版本，然后选择 **Tomcat/JWS** 作为后端配置。



注意

Red Hat JBoss Core Services 2.4.57 不支持将非升级的连接连接到后端 WebSocket 服务器。这意味着，当您为 **mod_proxy_wstunnel** 模块配置 **ProxyPass** 指令时，您必须确保 upgrade 参数没有设置为 **NONE**。有关 **mod_proxy_wstunnel** 的更多信息，请参阅 Apache 文档。

2.2.1. 将 Apache HTTP 服务器配置为加载 mod_jk

您可以通过在 **mod_jk.conf** 文件中指定配置设置，将 Apache HTTP 服务器配置为加载 **mod_jk**。根据您使用的 Apache HTTP 服务器分发，配置文件的位置会有所不同。

您还可以执行以下可选配置步骤：

- 除了 **JkMount** 指令外，您还可以使用 **JkMountFile** 指令来指定挂载点的配置文件。配置文件包含多个 Tomcat 转发的 URL 映射。
- 您可以配置作为负载均衡器功能的 Apache HTTP 服务器，以记录处理请求的每个 worker 节点的详细信息。如果您需要对负载均衡器进行故障排除，这非常有用。

先决条件

- 已安装 Apache HTTP 服务器。
- 如果您使用 Application Streams 安装 Apache HTTP 服务器的 RHEL 发行版，您已 手动安装 **mod_jk** 。

流程

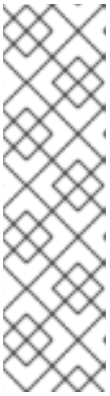
1. 进入 Apache HTTP 服务器配置目录：
   - 如果您使用 JBCS Apache HTTP 服务器，请转到 ***JBCS_HOME*/httpd/conf.d** 目录。
   - 如果您使用 RHEL Apache HTTP 服务器，请转到 **/etc/httpd/conf.d** 目录。
2. 创建名为 **mod_jk.conf** 的新文件，并输入以下配置详情：

```
# Load mod_jk module
# Specify the filename of the mod_jk lib
LoadModule jk_module modules/mod_jk.so

# Where to find workers.properties
JkWorkersFile conf.d/workers.properties

# Where to put jk logs
JkLogFile logs/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
JkLogStampFormat  "[%a %b %d %H:%M:%S %Y]"

# JkOptions indicates to send SSL KEY SIZE
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories
```

```
# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount your applications
JkMount /application/* loadbalancer

# Add shared memory.
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
    JkMount status
    Require ip 127.0.0.1
</Location>
```

重要

确保 **LoadModule** 指令引用您安装的 **mod_jk** 原生二进制文件。



注意

JkMount 指令指定 Apache HTTP 服务器可以转发到 **mod_jk** 模块的 URL。根据 **JkMount** 指令的配置，**mod_jk** 将收到的 URL 转发到正确的 servlet 容器。

要启用 Apache HTTP 服务器直接提供静态内容（或 PHP 内容），且只对 Java 应用程序使用负载均衡器，前面的配置示例指定，Apache HTTP 服务器仅向 **mod_jk** 负载均衡器使用 URL **/application/*** 发送请求。

另外，您还可以将 Apache HTTP 服务器配置为将所有 URL 转发到 **mod_jk**，方法是在 **JkMount** 指令中指定 /*。

3. 可选： 要使用 **JkMountFile** 指令为挂载点指定配置文件，请执行以下步骤：

   a. 进入 Apache HTTP 服务器配置目录：

      - 如果您使用 JBCS Apache HTTP 服务器，请转到 ***JBCS_HOME*/httpd/conf.d** 目录。
      - 如果您使用 RHEL Apache HTTP 服务器，请转到 **/etc/httpd/conf.d** 目录。

   b. 创建名为 **uriworkermap.properties** 的文件。

   c. 指定您要转发的 URL 和 worker 名称。
   例如：

```
# Simple worker configuration file

# Mount the Servlet context to the ajp13 worker
/application=loadbalancer
/application/*=loadbalancer
```

注意

所需的语法格式为：/***URL=WORKER_NAME***

前面的示例将 **mod_jk** 配置为将 **/application** 的请求转发到 JBoss Web Server Tomcat 后端。

d. 在 **mod_jk.conf** 文件中，输入以下指令：

```
# Use external file for mount points.
# It will be checked for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
JkMountFile conf.d/uriworkermap.properties
```

4. 可选： 要启用 Apache HTTP 服务器日志记录，请执行以下步骤之一：

- 在 **JkRequestLogFormat** 指令中包含 **%w**，如上一步中有关 **mod_jk.conf** 设置所示。
- 通过在 Apache HTTP Server **LogFormat**(s)中包含 **%{JK_WORKER_NAME}n**，记录您要使用的 **mod_jk** worker 的名称。

其他资源

- The Apache Tomcat Connectors - Web Server HowTo : mod_jk Directives
- Apache HTTP Server Documentation: Log Files

2.2.2. 在 mod_jk中配置 worker 节点

您可以通过在 worker **.properties** 文件中指定设置，将多个 worker 节点配置为处理 Apache HTTP 服务器转发到 servlet 容器的请求。根据您使用的 Apache HTTP 服务器分发，配置文件的位置会有所不同。

此流程中的示例演示了如何在加权循环配置中定义两个 **mod_jk** worker 节点，该配置使用两个 servlet 容器之间的粘性会话。

先决条件

- 您熟悉 **worker.properties** 指令的格式。
- 您已将 Apache HTTP 服务器配置为加载 **mod_jk**。

流程

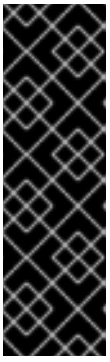
1. 进入 Apache HTTP 服务器配置目录：
   - 如果您使用 JBCS Apache HTTP 服务器，请转到 ***JBCS_HOME*/httpd/conf.d** 目录。
   - 如果您使用 RHEL Apache HTTP 服务器，请转到 **/etc/httpd/conf.d** 目录。
2. 创建名为 **workers.properties** 的文件。
3. 输入以下配置详情：

```
# Define list of workers that will be used
```

```
# for mapping requests
worker.list=loadbalancer,status

# Define Node1
# modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=node1.mydomain.com
worker.node1.type=ajp13
worker.node1.ping_mode=A
worker.node1.lbfactor=1
worker.node1.secret=<YourSecret>

# Define Node2
# modify the host as your host IP or DNS name.
worker.node2.port=8009
worker.node2.host=node2.mydomain.com
worker.node2.type=ajp13
worker.node2.ping_mode=A
worker.node2.lbfactor=1
worker.node1.secret=<YourSecret>

# Load-balancing behavior
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1

# Status worker for managing load balancer
worker.status.type=status
```

注意

在前面的示例中，确保将 **host**、**port** 和 **secret** 设置替换为与您的环境相关的值。



重要

使用 Tomcat AJP Connector 时，需要 **secret** 属性。您可以在 **workers.properties** 文件中指定 worker 节点或负载均衡器的 **secret** 属性。例如：

worker.*<WORKER_NAME>*.secret=*<YOUR_AJP_SECRET>*

在上例中，将 ***<WORKER_NAME>*** 和 ***<YOUR_AJP_SECRET>*** 替换为与您环境相关的值。

2.2.3. 配置 JBoss Web 服务器以使用 mod_jk

默认情况下，JBoss Web 服务器配置为接收来自 **mod_jk** 连接器的 Apache JServ 协议(AJP)流量。在 JBoss Web Server 主机上，在 ***JWS_HOME*/tomcat *<VERSION>* /conf/server.xml** 文件中默认配置 AJP 连接器。

但是，要使用带有 **mod_jk** 的 worker 节点，您必须执行以下附加配置步骤：

- 在 JBoss Web Server 主机上，在 **server.xml** 文件中，您必须在每个 worker 节点的 Engine 中为 **jvmRoute** 属性配置唯一值。

- 在 Apache HTTP Server 主机上，在 **worker.properties** 文件中，您必须为 worker 节点或负载均衡器指定 **secret** 属性。根据您使用的 Apache HTTP 服务器分发， **worker.properties** 文件的位置会有所不同。

注意

使用 Tomcat AJP 连接器时需要 **secret** 属性。

流程

1. 在 JBoss Web Server 主机上，为每个 worker 节点的 Engine 中为 **jvmRoute** 属性配置唯一值：
   a. 打开 ***JWS_HOME*/tomcat_<VERSION>_/conf/server.xml** 文件。
   b. 输入以下详情：

```
<Engine name="Catalina" jvmRoute="node1" >
```

重要

确保 **jvmRoute** 属性值与您在 Apache HTTP 服务器主机上 **worker.properties** 文件中指定的 worker 名称匹配。

2. 在 Apache HTTP Server 主机上，为 worker 节点或负载均衡器指定 **secret** 属性：
   a. 进入 Apache HTTP 服务器配置目录：
      - 如果您使用 JBCS Apache HTTP 服务器，请转到 ***JBCS_HOME*/httpd/conf.d** 目录。
      - 如果您使用 RHEL Apache HTTP 服务器，请转到 **/etc/httpd/conf.d** 目录。
   b. 打开 **workers.properties** 文件。
   c. 确保 **secret** 属性以以下格式指定：

```
worker.<WORKER_NAME>.secret=<YOUR_AJP_SECRET>`
```

注意

确保将 < ***WORKER_NAME>*** 和 < ***YOUR_AJP_SECRET*** > 替换为适合您环境的值。

注意

如果您使用 **ProxyPass** 指令在负载均衡器上设置 **secret**，负载均衡器的所有成员都会继承这个 **secret**。例如：

```
<Proxy balancer://mycluster>`
   BalancerMember ajp://node1:8009 route=node1
secret=YOUR_AJP_SECRET
   BalancerMember ajp://node2:8009 route=node2
secret=YOUR_AJP_SECRET
</Proxy>
ProxyPass /example/ balancer://mycluster/example/
stickysession=JSESSIONID|jsessionid
```

第 3 章 使用 JBOSS HTTP 连接器(MOD_PROXY_CLUSTER)的负载平衡

mod_proxy_cluster 连接器是一个减少配置、智能负载平衡解决方案，允许 Apache HTTP 服务器连接后端 JBoss Web 服务器或 JBoss EAP 主机。**mod_proxy_cluster** 模块基于最初开发 JBoss **mod_cluster** 社区项目的技术。

3.1. MOD_PROXY_CLUSTER 关键功能和组件

mod_proxy_cluster 模块对 JBoss EAP 和 JBoss Web Server worker 节点的 HTTP 请求进行负载均衡。**mod_proxy_cluster** 模块使用 Apache HTTP 服务器作为代理服务器。

mod_proxy_cluster的主要功能

mod_proxy_cluster 连接器比 **mod_jk** 连接器有几个优点：

- 启用 **mod_proxy_cluster** 模块后，**mod_proxy_cluster** 管理协议(MCMP)是 Tomcat 服务器和 Apache HTTP 服务器之间的额外连接。Tomcat 服务器使用 MCMP 通过使用自定义的 HTTP 方法将服务器端负载图和生命周期事件传输回 Apache HTTP 服务器。
- 使用 **mod_proxy_cluster** 的 Apache HTTP 服务器的动态配置允许带有 **mod_proxy_cluster** 侦听器的 Tomcat 服务器加入负载平衡安排，而无需手动配置。
- Tomcat 服务器执行负载计算，而不是依赖 Apache HTTP 服务器。这使得负载平衡指标比其他连接器更准确。
- **mod_proxy_cluster** 连接器提供精细的应用程序生命周期控制。每个 Tomcat 服务器将 Web 应用程序上下文生命周期事件转发到 Apache HTTP 服务器。这些生命周期事件包括通知 Apache HTTP 服务器为特定上下文启动或停止路由请求。这可防止最终用户因为资源不可用而看到 HTTP 错误。
- 您可以使用带有 **mod_proxy_cluster** 的 Apache JServ 协议(AJP)、Hypertext Transfer Protocol (HTTP)或 Hypertext Transfer Protocol Secure (HTTPS)传输。

mod_proxy_cluster 组件

在代理服务器中，**mod_proxy_cluster** 由四个 Apache 模块组成：

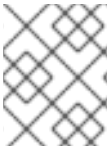
组件	Description
mod_cluster_slotmem.so	Shared Memory Manager 模块与多个 Apache HTTP 服务器进程共享实时 worker 节点信息。
mod_manager.so	Cluster Manager 模块接收和确认来自 worker 节点的信息，包括节点注册、节点加载数据和节点应用程序生命周期事件。
mod_proxy_cluster.so	Proxy Balancer 模块处理到集群节点的请求路由。Proxy Balancer 根据集群中的应用程序位置、每个集群节点的当前状态和会话 ID （如果请求是已建立的会话的一部分）选择适当的目标节点。
mod_advertise.so	Proxy Advertisement 模块通过 UDP 多播消息广播代理服务器是否存在。服务器公告消息包含代理服务器侦听要从要加入负载平衡集群的 worker 节点的 IP 地址和端口号。

其他资源

- **mod_proxy_cluster** 连接器模块

3.2. MOD_PROXY_CLUSTER 安装和升级

红帽 JBoss 核心服务(JBCS)和 Red Hat Enterprise Linux (RHEL)提供单独的 Apache HTTP 服务器分发。您安装的 Apache HTTP 服务器分发决定了 **mod_proxy_cluster** 连接器的安装是自动的，还是需要手动步骤。根据您安装的 Apache HTTP 服务器的发布，**mod_proxy_cluster** 模块和配置文件的安装路径也会不同。



注意

JBCS Apache HTTP 服务器支持在所有支持的操作系统中使用 **mod_proxy_cluster**。
RHEL Apache HTTP 服务器只支持在 RHEL 9 上使用 **mod_proxy_cluster**。

3.2.1. 使用 JBCS Apache HTTP 服务器安装 mod_proxy_cluster

JBCS 安装的 Apache HTTP 服务器部分会自动安装 **mod_proxy_cluster** 模块。

您可以按照 Red Hat JBoss Core Services Apache HTTP Server 安装指南中的步骤为您的操作系统安装或升级到最新的 JBCS Apache HTTP 服务器发行版本。如需更多信息，**请参阅附加资源** 链接。

在使用 JBCS Apache HTTP 服务器时，请考虑以下有关 **mod_proxy_cluster** 安装的准则：

- **mod_proxy_cluster.so**,**mod_cluster_slotmem.so**,**mod_manager.so**, 和 **mod_advertise.so** 模块安装在 ***JBCS_HOME*/httpd/modules** 目录中。
- **mod_proxy_cluster.conf.sample** 配置文件位于 ***JBCS_HOME*/httpd/conf.d** 目录中。
- **mod_proxy_cluster.conf.sample** 文件包含 **mod_proxy_cluster** 模块的 **LoadModule** 指令。

注意

JBCS_HOME 代表 JBCS 安装的顶级目录，即 **/opt/jbcs-httpd24-2.4**。

其他资源

- 从归档文件在 RHEL 上安装 JBCS Apache HTTP 服务器
- 从 RPM 软件包在 RHEL 7 或 RHEL 8 上安装 JBCS Apache HTTP 服务器
- 在 Windows Server 上安装 JBCS Apache HTTP 服务器

3.2.2. 从以前的 JBCS 发行版本升级 mod_proxy_cluster

2.4.37 及更早的版本中提供的 **mod_cluster-native** 软件包在 JBCS 2.4.51 或更高版本中重命名了 **mod_proxy_cluster**。作为此更改的一部分，2.4.37 及更早的版本中提供的 **mod_cluster.conf** 文件也会在 JBCS 2.4.51 或更高版本中重命名了 **mod_proxy_cluster.conf**。JBCS 以不同的方式处理现有 **mod_proxy_cluster** 配置的升级，具体取决于您从归档文件或 RPM 软件包安装 JBCS。

从 RPM 软件包安装时，升级 mod_proxy_cluster 配置
如果要升级从 RHEL 7 或 RHEL 8 上的 RPM 软件包安装的现有 JBCS 安装，请考虑以下指南：

- 如果您要从 JBCS 2.4.37 或更早版本升级，则 JBCS 会在升级过程中保留现有的 **mod_cluster.conf** 文件。在这种情况下，升级的 JBCS 2.4.57 部署包括您现有的 **mod_cluster.conf** 文件和默认的 **mod_proxy_cluster.conf** 文件。如果您随后希望迁移到使用 **mod_proxy_cluster.conf**，您可以手动更新默认的 **mod_proxy_cluster.conf** 文件，以满足您的设置要求。

- 如果您要从 JBCS 2.4.51 升级，则 JBCS 会在升级过程中保留现有的 **mod_proxy_cluster.conf** 文件。在这种情况下，升级的 JBCS 2.4.57 部署包括您现有的 **mod_proxy_cluster.conf** 文件和默认的 **mod_proxy_cluster.conf.rpmnew** 文件。

从存档文件安装时，升级 mod_proxy_cluster 配置
如果您要升级从存档文件安装的现有 JBCS 安装，请考虑以下准则：

- 如果您要从 JBCS 2.4.37 或更早版本升级，则不需要对提取 2.4.57 归档文件执行任何操作。JBCS 2.4.57 不包含默认的 **mod_cluster.conf** 文件，因此您现有的 **mod_cluster.conf** 文件会在产品升级过程中保留。在这种情况下，升级的 JBCS 2.4.57 部署包括您现有的 **mod_cluster.conf** 文件和默认的 **mod_proxy_cluster.conf** 文件。如果您随后希望迁移到使用 **mod_proxy_cluster.conf**，您可以手动更新默认的 **mod_proxy_cluster.conf** 文件，以满足您的设置要求。

- 如果您要从 JBCS 2.4.51 或 JBCS 2.4.57 的现有发行版本升级，您必须首先将现有 **mod_proxy_cluster.conf** 文件复制到临时位置。JBCS 2.4.57 包含默认的 **mod_proxy_cluster.conf** 文件，该文件会在产品升级过程中自动覆盖现有的 **mod_proxy_cluster.conf** 文件。提取最新的 2.4.57 归档文件后，您可以将现有 **mod_proxy_cluster.conf** 文件的备份复制到正确的位置，以覆盖默认文件。

3.2.3. 使用 RHEL Application Streams 安装 mod_proxy_cluster

如果您使用 Application Streams 从 RPM 软件包安装 Apache HTTP 服务器的 RHEL 9 发行版，RHEL 不会自动安装 **mod_proxy_cluster** 软件包。在这种情况下，如果要使用 **mod_proxy_cluster** 连接器，您必须手动安装 **mod_proxy_cluster** 软件包。

先决条件

- 已使用 Application Streams 在 RHEL 9 上安装了 Apache HTTP 服务器。

流程

- 以 root 用户身份输入以下命令：

```
# dnf install mod_proxy_cluster
```

验证

- 要检查 **mod_proxy_cluster** 软件包是否已成功安装，请输入以下命令：

```
# rpm -q mod_proxy_cluster
```

  前面的命令输出已安装软件包的全名，其中包括版本和平台信息。

在使用 RHEL Application Streams 时，请考虑以下有关 **mod_proxy_cluster** 安装的指南：

- **mod_proxy_cluster.so**,**mod_cluster_slotmem.so**,**mod_manager.so**, 和 **mod_advertise.so** 模块安装在 **/usr/lib64/httpd/modules** 目录中。

- **mod_proxy_cluster.conf.sample** 配置文件位于 **/etc/httpd/conf.d** 目录中。
- **mod_proxy_cluster.conf.sample** 文件包含 **mod_proxy_cluster** 模块的 **LoadModule** 指令。

其他资源

- 应用程序流
- 使用 DNF 工具管理软件

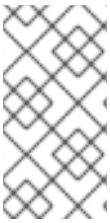
3.3. 使用 MOD_PROXY_CLUSTER时 APACHE HTTP 服务器负载均衡配置

在 Apache HTTP Server 2.1 及更高版本中，默认情况下，Apache HTTP 服务器正确配置了 **mod_proxy_cluster**。有关设置自定义配置的更多信息，请参阅配置基本代理服务器。

mod_proxy_cluster的配置文件示例
根据您是否通过 Red Hat JBoss Core Services (JBCS)或使用 Red Hat Enterprise Linux (RHEL) Application Streams 安装 **mod_proxy_cluster**，请考虑以下指南：

- JBCS 在 ***JBCS_HOME*/httpd/conf.d/ 目录中提供了 mod_ proxy_cluster** 的示例配置文件。
- RHEL 在 **/etc/httpd/conf.d/** 目录中提供了 **mod_proxy_cluster** 的示例配置文件。

mod_proxy_cluster 的示例配置文件命名为 **mod_proxy_cluster.conf.sample**。要使用此示例而不是创建自己的配置文件，您可以删除 **.sample** 扩展，并根据需要修改文件内容。

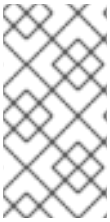


注意

您还可以使用红帽客户门户网站上的 Load Balancer 配置工具，快速为 **mod_proxy_cluster** 和 Tomcat worker 节点生成最佳配置模板。当您将 Load Balancer Configuration 工具用于 Apache HTTP Server 2.4.57 时，请确保选择 **2.4.x** 作为 Apache 版本，然后选择 **Tomcat/JWS** 作为后端配置。

使用 mod_proxy_cluster的指南
考虑以下使用 **mod_proxy_cluster** 连接器的准则：

- 当使用 **mod_proxy_cluster** 连接器时，您必须启用 **mod_proxy** 模块并禁用 **mod_proxy_balancer** 模块。
- 如果您希望 **mod_proxy_cluster** 使用 Apache JServ 协议(AJP)，您必须启用 **proxy_ajp_module**。
- 使用 AJPSecret **your_secret** 为 AJP 后端提供机密。如果 **your_secret** 与后端中配置的值不匹配，则后端会为通过代理发送的任何请求发送 **503** 错误响应。



注意

Red Hat JBoss Core Services 2.4.57 不支持将非升级的连接连接到后端 websocket 服务器。这意味着，当您为 **mod_proxy_wstunnel** 模块配置 **ProxyPass** 指令时，您必须确保 upgrade 参数没有设置为 **NONE**。有关 **mod_proxy_wstunnel** 的更多信息，请参阅 Apache 文档。

3.3.1. 配置基本代理服务器

您可以将 Apache HTTP 服务器配置为充当代理服务器，以转发 web 客户端和后端 web 服务器之间的请求和响应。您必须配置代理服务器监听程序，以接收来自后端 worker 节点的连接请求和响应。当您要配置使用 **mod_proxy_cluster** 的负载平衡代理服务器时，还必须为管理频道配置虚拟主机。

先决条件

- 已安装 Apache HTTP 服务器。
- 如果您使用 Application Streams 安装 Apache HTTP 服务器的 RHEL 发行版，您可以 手动安装 **mod_proxy_cluster** 。
- 您必须为代理服务器监听程序指定的端口用于传入的 TCP 连接。

流程

1. 进入 Apache HTTP 服务器配置目录：
   - 如果您使用 JBCS Apache HTTP 服务器，请转到 ***JBCS_HOME*/httpd/conf.d** 目录。
   - 如果您使用 RHEL Apache HTTP 服务器，请转到 **/etc/httpd/conf.d** 目录。
2. 打开 **mod_proxy_cluster.conf** 文件。
3. 要为代理服务器创建 **Listen** 指令，请在 **mod_proxy_cluster.conf** 文件中输入以下行：

```
Listen IP_ADDRESS:PORT_NUMBER
```

注意

在前面的示例中，将 ***IP_ADDRESS*** 替换为代理服务器用来与 worker 节点通信的服务器网络接口地址，并将 ***PORT_NUMBER*** 替换为代理服务器侦听的端口。

确保端口为传入 TCP 连接打开。

4. 要创建虚拟主机，请在 **mod_proxy_cluster.conf** 文件中输入以下详情：

```
<VirtualHost IP_ADDRESS:PORT_NUMBER>

  <Directory />
    Require ip IP_ADDRESS
  </Directory>

  KeepAliveTimeout 60
  MaxKeepAliveRequests 0

  ManagerBalancerName mycluster
  AdvertiseFrequency 5
  EnableMCPMReceive On

</VirtualHost>
```

注意

在前面的示例中，将 ***IP_ADDRESS*** 和 ***PORT_NUMBER*** 替换为您为 **Listen** 指令指定的服务器网络接口地址和端口号。

此地址和端口组合仅用于 **mod_proxy_cluster** 管理消息。此地址和端口组合不用于常规流量。

有关启动 Apache HTTP 服务器服务的更多信息，请参阅 Red Hat JBoss Core Services Apache HTTP Server 安装指南。

3.3.1.1. 禁用服务器广告

代理服务器使用 UDP 多播来公告其自身。**AdvertiseFrequency** 指令指示服务器默认每 10 秒发送服务器广告消息。服务器广告消息包含您在 **VirtualHost** 定义中指定的 ***IP_ADDRESS*** 和 ***PORT_NUMBER***。配置为响应服务器广告的 worker 节点使用此信息将自身注册到代理服务器。如果要防止 worker 节点使用代理服务器注册，您可以选择禁用服务器广告。

注意

当代理服务器和 worker 节点之间有 UDP 多播时，服务器广告会添加 worker 节点，而无需在代理服务器上进一步配置。服务器广告只需要 worker 节点上的配置最小。

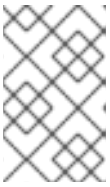
先决条件

- 您已配置了基本代理服务器。

流程

1. 进入 Apache HTTP 服务器配置目录：
   - 如果您使用 JBCS Apache HTTP 服务器，请转到 ***JBCS_HOME*/httpd/conf.d** 目录。
   - 如果您使用 RHEL Apache HTTP 服务器，请转到 **/etc/httpd/conf.d** 目录。
2. 打开 **mod_proxy_cluster.conf** 文件。
3. 在 **VirtualHost** 定义中添加以下指令：

```
ServerAdvertise Off
```



注意

如果禁用服务器公告，或者代理服务器和 worker 节点之间没有 UDP 多播，您可以使用静态代理服务器列表配置 worker 节点。在这两种情况下，您不需要使用 worker 节点列表配置代理服务器。

其他资源

- 使用静态代理服务器列表配置 worker 节点

3.3.1.2. 日志记录 worker 节点详情

当您配置使用 **mod_proxy_cluster** 的负载平衡代理服务器时，您可以选择将 Apache HTTP 服务器配置为记录处理请求的每个 worker 节点的详细信息。如果您需要对负载均衡器进行故障排除，日志记录 worker 节点详情会很有用。

先决条件

- 您已配置了基本代理服务器。

流程

1. 进入 Apache HTTP 服务器配置目录：
   - 如果您使用 JBCS Apache HTTP 服务器，请转到 ***JBCS_HOME*/httpd/conf.d** 目录。
   - 如果您使用 RHEL Apache HTTP 服务器，请转到 **/etc/httpd/conf.d** 目录。
2. 打开 **mod_proxy_cluster.conf** 文件。
3. 在 Apache HTTP Server **LogFormat** 指令中添加以下详情：

```
%{BALANCER_NAME}e ::
The name of the balancer that served the request.

%{BALANCER_WORKER_NAME}e ::
The name of the worker node that served the request.
```

其他资源

- 有关日志文件的 Apache HTTP 服务器文档

3.3.2. 在 mod_proxy_cluster中配置 JBoss Web Server worker 节点

使用 **mod_proxy_cluster** 时，您可以将后端 worker 节点配置为仅在非集群模式下运行的 JBoss Web Server Tomcat 服务。在这种情况下，**mod_proxy_cluster** 在计算负载均衡因素时，在任何特定时间都只能使用一个负载指标。

注意

JBoss Web Server worker 节点只支持 **mod_proxy_cluster** 功能的子集。JBoss EAP 提供了完整的 **mod_proxy_cluster** 功能。

先决条件

- 熟悉 **mod_proxy_cluster**的代理和代理发现配置属性。

流程

1. 要为 JBoss Web 服务器添加监听程序，在 ***JWS_HOME*/tomcat <*VERSION*> /conf/server.xml** 文件中添加以下 **Listener** 元素，在其他 **Listener** 元素下：

```
<Listener
className="org.jboss.modcluster.container.catalina.standalone.ModClusterListener"
advertise="true" stickySession="true" stickySessionForce="false"
stickySessionRemove="true" />
```

2. 要为 worker 节点赋予一个唯一的身份，在 ***JWS_HOME*/tomcat <*VERSION*>/conf/server.xml** 文件中，将 **jvmRoute** 属性和值添加到 **Engine** 元素：

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="worker01">
```

3. 要配置 **STATUS MCMP** 消息频率，请修改 **org.jboss.modcluster.container.catalina.status-frequency** Java 系统属性。
例如：

```
-Dorg.jboss.modcluster.container.catalina.status-frequency=6
```

注意

JBoss Web Server worker 节点定期向 Apache HTTP 服务器平衡发送包含其当前负载状态的状态消息。这些消息的默认频率为 10 秒。如果您有数百个 worker 节点，S**TAMP** 消息可以增加 Apache HTTP 服务器网络上的流量拥塞。

您可以通过修改 **org.jboss.modcluster.container.catalina.status-frequency** Java 系统属性来配置 **MCMP** 消息频率。默认情况下，属性接受以秒为单位指定的值，乘以 10。例如，将属性设为 **1** 表示 10 秒。在前面的示例中，属性设置为 **6**，即 60 秒。

4. 可选：要为代理服务器公告配置防火墙，请完成以下步骤，以便在 worker 节点的防火墙上为 UDP 连接打开端口 **23364** ：

- 对于 RHEL：

```
firewall-cmd --permanent --zone=public --add-port=23364/udp
```

- 对于使用 PowerShell 的 Windows Server：

```
Start-Process "$psHome\powershell.exe" -Verb Runas -ArgumentList '-command "NetSh Advfirewall firewall add rule name="UDP Port 23364" dir=in  action=allow protocol=UDP localport=23364"'
Start-Process "$psHome\powershell.exe" -Verb Runas -ArgumentList '-command "NetSh Advfirewall firewall add rule name="UDP Port 23364" dir=out action=allow protocol=UDP localport=23364"'
```

注意

当代理服务器使用 **mod_proxy_cluster** 时，代理服务器可以使用 UDP 多播来公告其自身。大多数操作系统防火墙默认阻止服务器公告功能。要启用服务器公告并接收这些多播消息，您可以为 worker 节点的防火墙上的 UDP 连接打开端口 **23364**，如上例中所示。

3.3.3. 配置 worker 节点以使用静态代理服务器列表操作

服务器公告允许 worker 节点动态发现并使用代理服务器注册。如果 UDP 多播不可用，或者禁用服务器广告，则必须使用代理服务器地址和端口的静态列表配置 JBoss Web Server worker 节点。

先决条件

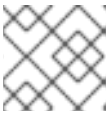
- 您已配置了 JBoss Web Server worker 节点。

- 熟悉 Tomcat 的代理配置参数。

流程

1. 打开 ***JWS_HOME*/tomcat <*VERSION*> /conf/server.xml** 文件。
2. 要定义 **mod_proxy_cluster** 侦听器并禁用动态代理发现，修改 **ModClusterListener** 的 **Listener** 元素。
   例如：

```
<Listener
className="org.jboss.modcluster.container.catalina.standalone.ModClusterListener"
advertise="false" stickySession="true" stickySessionForce="false"
stickySessionRemove="true"/>
```



注意

确保将 **advertise** 属性设置为 **false**。

3. 要创建静态代理服务器列表，请通过以以下格式添加以逗号分隔的代理列表来更新 **proxyList** 属性：*IP_ADDRESS*: PORT , ***IP_ADDRESS*:*PORT*:*PORT***
   例如：

```
<Listener
className="org.jboss.modcluster.container.catalina.standalone.ModClusterListener"
advertise="false" stickySession="true" stickySessionForce="false"
stickySessionRemove="true" proxyList="10.33.144.3:6666,10.33.144.1:6666"/>
```

3.4. MOD_PROXY_CLUSTER 字符限制

mod_proxy_cluster 模块使用共享内存来保持节点描述。共享内存在 Apache HTTP 服务器启动时创建，每个项目的结构已被修复。

当您定义代理服务器和 worker 节点属性时，请确保遵循以下字符限制：

属性	最大字符限制	Description
别名长度	100 个字符	alias 对应于对应虚拟主机的网络名称；名称在 **Host** 元素中定义。
上下文长度	40 个字符	例如，如果 **myapp.war** 部署在 **/myapp** 中，则上下文中包含 **/myapp**。
负载均衡器名称长度	40 个字符	这是 <**Listener> 元素中 的负载均衡器**。
jvmRoute 字符串长度	80 个字符	**<Engine>** 元素中的 **JVMRoute**。

属性	最大字符限制	Description
域名长度	20 个字符	这是 < **Listener** > 元素中的 **loadBalancingGroup**。
节点的主机名长度	64 个字符	这是 < **Connector>** 元素中的主机名地址。
节点的端口长度	7 个字符	这是 < **Connector** > 元素中的 port 属性。例如：**8009** 为 4 个字符。
节点的方案长度	6 个字符	这是连接器的协议。可能的值有 **http**、**https** 和 **ajp**。
cookie 名称长度	30 个字符	这是会话 ID 的标头 Cookie 名称。默认值为 **JSESSIONID**，它基于 **org.apache.catalina.Globals.SESSION_COOKIE_NAME** 属性。
路径名称长度	30 个字符	这是会话 ID 的参数名称。默认值为 **JSESSIONID**，它基于 **org.apache.catalina.Globals.SESSION_PARAMETER_NAME** 属性。
会话 ID 长度	120 个字符	会话 ID 采用以下格式：**BE81FAA969BF64C8EC2B6600457EAAAA.node01**

第 4 章 使用 MOD_PROXY_CLUSTER的负载均衡配置示例

您可以将 JBCS 配置为使用 **mod_proxy_cluster** 连接器在 Red Hat Enterprise Linux 系统中进行负载平衡。

当您要配置使用 **mod_proxy_cluster** 的负载平衡解决方案时，您必须执行以下任务：

1. 将 JBCS 设置为代理服务器。
2. 配置 Tomcat worker 节点。
3. 定义 iptables 防火墙规则。

4.1. 将 JBCS 设置为代理服务器

当您将 JBCS 配置为使用 **mod_proxy_cluster** 时，您必须通过在 **mod_proxy_cluster.conf** 文件中指定配置详情将 JBCS 设置为代理服务器。

流程

1. 进入 ***JBCS_HOME*/httpd/conf.d/** 目录。
2. 创建名为 **mod_proxy_cluster.conf** 的文件。
3. 输入以下配置详情：

```
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule cluster_slotmem_module modules/mod_cluster_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule advertise_module modules/mod_advertise.so

MemManagerFile cache/mod_proxy_cluster

<IfModule manager_module>
  Listen 6666
  <VirtualHost *:6666>
    <Directory />
      Require ip 127.0.0.1
    </Directory>
    ServerAdvertise on
    EnableMCPMReceive
    <Location /mod_cluster_manager>
      SetHandler mod_cluster-manager
      Require ip 127.0.0.1
   </Location>
  </VirtualHost>
</IfModule>
```

重要

如上例所示，**mod_proxy_cluster** 软件包要求您将 **conf.d** 文件中的 **MemManagerFile** 指令设置为 **cache/mod_proxy_cluster**。

注意

前面的示例演示了如何将 JBCS 设置为侦听 **localhost** 的代理服务器。

4.2. 配置 TOMCAT WORKER 节点

当您将 JBCS 配置为使用 **mod_proxy_cluster** 时，您必须通过将 **Listener** 元素添加到 **server.xml** 文件来配置 Tomcat worker 节点。

先决条件

- 您已将 JBCS 设置为代理服务器。

流程

1. 打开 ***JWS_HOME*/tomcat <*VERSION*> /conf/server.xml** 文件。
2. 添加以下 **Listener** 元素：

```
<Listener
className="org.jboss.modcluster.container.catalina.standalone.ModClusterListener"
advertise="true"/>
```

4.3. 定义 IPTABLES 防火墙规则示例

当您将 JBCS 配置为使用 **mod_proxy_cluster** 时，您必须使用 **iptables** 定义防火墙规则。

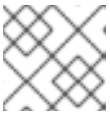
先决条件

- 您已配置了 Tomcat worker 节点。

流程

- 使用 **iptables** 定义一组防火墙规则。
  例如：

```
/sbin/iptables -I INPUT 5 -p udp -d 224.0.1.0/24 -j ACCEPT -m comment --comment
"mod_proxy_cluster traffic"
/sbin/iptables -I INPUT 6 -p udp -d 224.0.0.0/4 -j ACCEPT -m comment --comment "JBoss
Cluster traffic"
/sbin/iptables -I INPUT 9 -p udp -s 192.168.1.0/24 -j ACCEPT -m comment --comment
"cluster subnet for inter-node communication"
/sbin/iptables -I INPUT 10 -p tcp -s 192.168.1.0/24 -j ACCEPT -m comment --comment
"cluster subnet for inter-node communication"
/etc/init.d/iptables save
```



注意

前面的示例演示了为 **192.168.1.0/24** 子网上的集群节点定义防火墙规则。

附录 A. MOD_PROXY 连接器模块

mod_proxy 连接器包含一组标准 Apache HTTP 服务器模块。这些模块使 Apache HTTP 服务器能够充当代理/网关，用于通过不同类型的协议在 Web 客户端和后端服务器之间发送 Web 流量。

本附录描述了 **mod_proxy** 连接器使用的模块。

A.1. MOD_PROXY.SO MODULE

mod_proxy.so 模块是一种标准的 Apache HTTP Server 模块，使服务器能够充当通过 AJP (Apache JServe 协议)、FTP、CONNECT （对于 SSL）传输的数据的代理。**mod_proxy** 模块不需要额外的配置。**mod_proxy** 模块的标识符是 **proxy_module**。

其他资源

- Apache 模块 mod_proxy

A.2. MOD_PROXY_AJP.SO MODULE

mod_proxy_ajp.so 模块是标准的 Apache HTTP 服务器模块，为 Apache JServ 协议(AJP)代理提供支持。通过使用 **mod_proxy_ajp** 模块，Apache HTTP 服务器充当在 Web 客户端和后端服务器之间发送 AJP 请求和响应的中间人。AJP 是一种明文协议，不支持数据加密。

如果要使用 **mod_proxy _ ajp，则需要 mod_proxy** 模块。**mod_proxy_ajp** 模块的标识符是 **proxy_ajp_module**。

此外，在使用 Tomcat AJP Connector 时，需要 **secret** 属性。您可以使用以下命令将 **secret** 属性添加到 **ProxyPass** 设置中：

ProxyPass /example/ ajp://localhost:8009/example/ secret=YOUR_AJP_SECRET

注意

如果在负载均衡器中设置了 **secret**，则它的所有成员都会继承这个 **secret**。

mod_proxy_ajp 模块不提供任何配置指令。

其他资源

- Apache Module mod_proxy_ajp

A.3. MOD_PROXY_HTTP.SO 模块

mod_proxy_http.so 模块是标准的 Apache HTTP 服务器模块，它支持 Hypertext 传输协议(HTTP)和 Hypertext 传输协议安全(HTTPS)代理。通过使用 **mod_proxy_http** 模块，Apache HTTP 服务器充当在 web 客户端和后端服务器之间转发 HTTP 或 HTTPS 请求的中间人。**mod_proxy_http** 模块支持 HTTP/1.1 和较早版本的 HTTP 协议。

如果要使用 **mod_proxy _ http，则需要 mod_proxy** 模块。**mod_proxy_http** 模块的标识符是 **proxy_http_module**。

mod_proxy_http 模块不提供任何配置指令。除了控制 **mod_proxy** 模块行为的配置外，**mod_proxy_http** 模块使用一系列控制 HTTP 协议提供程序行为的环境变量。https://httpd.apache.org/docs/2.4/mod/mod_proxy_http.html#env

其他资源

- Apache 模块 mod_proxy_http

A.4. MOD_PROXY_HTTP2.SO 模块

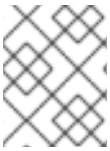
mod_proxy_http2.so 模块是标准的 Apache HTTP 服务器模块，支持 Hypertext 传输协议 2.0 (HTTP/2) 代理。通过使用 **mod_proxy_http2** 模块，Apache HTTP 服务器充当在 web 客户端和后端服务器之间转发 HTTP/2 请求的中间人。

mod_proxy_http2 模块支持使用 HTTP/1.1 或 HTTP/2 作为通信协议的客户端请求。但是，**mod_proxy_http2** 模块要求 Apache HTTP 服务器和后端服务器之间的所有通信都仅使用 HTTP/2。

对于具有相同后端目的地的客户端请求，Apache HTTP 服务器会尽可能重复使用相同的 TCP 连接。但是，即使您想将多个客户端请求转发到同一后端，Apache HTTP 服务器会为每个 HTTP/1.1 客户端请求转发一个单独的 HTTP/2 代理请求。

如果要使用 **mod_proxy _ http2，还需要 mod_proxy** 模块。**mod_proxy_http2** 模块的标识符是 **proxy_http2_module**。

mod_proxy_http2 模块不提供任何配置指令。



注意

mod_proxy_http2 模块是一种实验性 Apache 功能，要求将 **libnghttp2** 库用于核心 HTTP/2 引擎。

其他资源

- 为 JBCS Apache HTTP 服务器启用 HTTP/2
- Apache 模块 mod_proxy_http2

附录 B. MOD_JK CONNECTOR 模块

Apache Tomcat Connector, **mod_jk** 是 Apache Tomcat 项目提供的 Web 服务器插件。Apache HTTP 服务器可以使用 **mod_jk** 模块将 HTTP 客户端请求负载平衡到后端 servlet 容器，同时维护粘性会话并通过 Apache JServ 协议(AJP)进行通信。**mod_jk** 模块包含在 JBoss 核心服务安装的 Apache HTTP 服务器部分。

mod_jk 模块要求您在 Apache HTTP 服务器主机上创建 **mod_jk.conf** 文件和 **workers.properties** 文件。**mod_jk.conf** 文件指定要加载和配置 **mod_jk.so** 模块的设置。**worker.properties** 文件指定后端 worker 节点详情。您还必须在 [JWSShortName] 主机上配置一些设置以启用 **mod_jk** 支持。

其他资源

- 使用 Apache Tomcat Connector 进行负载平衡(**mod_jk**)
- **mod_jk**的 **worker.properties** 文件

附录 C. MOD_PROXY_CLUSTER 连接器模块

mod_proxy_cluster 连接器是基于最初开发的 JBoss mod_cluster 社区项目的技术减少了配置智能负载平衡解决方案。**mod_proxy_cluster** 连接器使 Apache HTTP 服务器充当高级负载平衡器，用于将流量转发到在 JBoss Web 服务器或 JBoss EAP 主机上运行的后端应用。**mod_proxy_cluster** 连接器提供 **mod_jk** 的所有功能，以及实时负载平衡计算、应用生命周期控制、自动代理发现和多个协议支持等功能。

本附录描述了 **mod_proxy_cluster** 连接器使用的模块。

注意

您可以使用 **mod_proxy** 的可配置指令（如 **ProxyIOBufferSize** ）配置 **mod_proxy_cluster** 连接器。

C.1. MOD_MANAGER.SO 模块和指令

集群管理器模块 **mod_manager.so** 接收来自节点的信息，包括 worker 节点注册、worker 节点加载数据和 worker 节点应用程序生命周期事件。

```
LoadModule manager_module modules/mod_manager.so
```

mod_manager.so的可配置指令

< **VirtualHost>** 元素中的可配置指令如下：

EnableMCPMReceive

允许 **VirtualHost** 接收 mod_cluster 管理协议(MCMP)消息。在 Apache HTTP 服务器配置中添加一个 **EnableMCPMReceive** 指令，以允许 **mod_proxy_cluster** 正确运行。**EnableMCPMReceive** 必须添加到配置 **advertise** 的位置的 **VirtualHost** 配置中。

MaxMCMPMaxMessSize

定义 MCMP 消息的最大大小。默认值从其他 **Max** 指令计算。最小值为 **1024**。

AllowDisplay

在 **mod_cluster-manager** 主页面中切换额外的显示。默认值为 **off**，这会导致在 **mod_cluster-manager** 主页上仅显示版本信息。

AllowCmd

使用 **mod_cluster-manager** URL 切换命令的权限。默认值为 **on**，允许命令。

ReduceDisplay

切换 **mod_cluster-manager** 页面中显示的信息缩减。减少信息允许在页面中显示更多节点。默认值为 **off**，它允许显示所有可用的信息。

MemManagerFile

定义 mod_manager 存储配置详情的文件的位置。mod_manager 还将此位置用于共享内存和锁定文件。这**必须是绝对路径名**。建议该路径位于本地驱动器中，而不是 NFS 共享。默认值为 **/logs/**。

Maxcontext

mod_proxy_cluster 将使用的最大上下文数。默认值为 **100**。

Maxnode

mod_proxy_cluster 将要使用的 worker 节点的最大数量。默认值为 **20**。

Maxhost

mod_proxy_cluster 将使用的最大主机数（别名）。这也是负载均衡器的最大数量。默认值为 **20**。

Maxsessionid

存储的活动会话标识符的最大数量。当没有从该会话收到信息五分钟时，会话被视为不活跃。这仅用于演示和调试目的。默认值为 **0，** 它会禁用此逻辑。

ManagerBalancerName

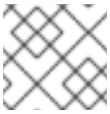
worker 节点不提供负载均衡器名称时使用的负载均衡器名称。默认值为 **mycluster**。

PersistSlots

在 **上** 设置为 时，节点、别名和上下文将在文件中保留。默认值为 **off**。

CheckNonce

在 **上** 设置为 时，将检查会话标识符，以确保它们是唯一的且之前尚未发生。默认值为 **on**。



注意

将此指令设置为 **off** 可能会使服务器易受重播攻击。

SetHandler mod_cluster-manager

定义显示集群中 worker 节点信息的处理程序。这在 **Location** 元素中定义：

```
<Location $LOCATION>
  SetHandler mod_cluster-manager
  Require ip 127.0.0.1
</Location>
```

在这种情况下，**$LOCATION** 也定义为 **mod_cluster_manager**。

在浏览器的 **Location** 元素中定义的 **$LOCATION** 时，请考虑以下准则：

- **传输** 对应于发送到 worker 节点的 POST 数据。
- **连接** 对应于请求此状态页面时已处理的请求数。
- **会话** 对应于活动会话的数量。**Maxsessionid** 为 **0** 时，不会出现此字段。

C.2. MOD_PROXY_CLUSTER.SO 模块和指令

Proxy Balancer 模块 **mod_proxy_cluster.so** 处理请求到集群节点的路由。Proxy Balancer 根据集群中的应用程序位置、每个集群节点的当前状态和会话 ID （如果请求是已建立的会话的一部分），选择适当的节点来转发请求。

```
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
```

mod_proxy_cluster.so的可配置指令

您还可以在 < **VirtualHost> 元素中配置** 以下指令，以更改负载均衡行为。

CreateBalancers

定义如何在 Apache HTTP 服务器虚拟主机中创建负载均衡器。以下值在 **CreateBalancers** 中有效：

- **0** ：在 Apache HTTP 服务器中定义的所有虚拟主机中创建负载均衡器。记得在 **ProxyPass** 指令中配置负载均衡器。

- **1**：不要创建负载均衡器。使用这个值时，还必须在 **ProxyPass** 或 **ProxyPassMatch** 中定义负载均衡器名称。
- **2**：仅创建主服务器。这是 **CreateBalancers** 的默认值。

UseAlias

定义是否检查定义的 **Alias** 对应于 **ServerName**。以下值是 **UseAlias** 的有效值：

- **0** ：忽略来自 worker 节点的别名信息。这是 **UseAlias** 的默认值。
- **1**：验证定义的别名是否与 worker 节点的服务器名称对应。

LBstatusRecalTime

定义代理计算 worker 节点状态之间的间隔（以秒为单位）。默认间隔为 **5** 秒。

ProxyPassMatch; ProxyPass

ProxyPass 将远程服务器映射到本地服务器命名空间中。如果本地服务器有地址，如 **http://local.com/**，以下 **ProxyPass** 指令会将 **http://local.com/requested/file1** 的本地请求转换为 **http://worker.local.com/file1** 的代理请求。

```
ProxyPass /requested/ http://worker.local.com/
```

ProxyPassMatch 使用正则表达式匹配代理 URL 应该应用到的本地路径。

对于任一指令，**!** 表示指定的路径为本地，并且该路径的请求不应路由到远程服务器。例如，以下指令指定应在本地提供 **gif** 文件。

```
ProxyPassMatch ^(/.*\.gif)$ !
```

UseNocanon

定义是否在不修改的情况下将原始 URL 路径转发到后端。
默认值为 **Off**。当 **UseNocanon** 指令设置为 **Off** 时，代理可将修改的 URL 转发到后端。但是，如果后端应用需要客户端请求的原始 URL 路径，则修改后的 URL 路径可能会导致意外问题。

当您将 **UseNocanon** 指令设置为 **On** 时，代理可以在不需要任何修改的情况下将原始 URL 路径转发到后端。在这种情况下，代理行为取决于您是否也在 **mod_proxy_cluster.conf** 文件中为请求的 URL 定义上下文和 **ProxyPass** 指令。上下文也称为 *虚拟主机定义*。

当您将 **UseNocanon** 指令设置为 **On** 时，请考虑以下准则：

- 如果您为请求的 URL 定义上下文，但没有为这个 URL 定义 **ProxyPass** 指令，代理将使用 **UseNocanon** 指令。
- 如果您为请求的 URL 定义上下文和 **ProxyPass** 指令，并且 **ProxyPass** 指令包含 **nocanon** 标志，代理使用 **nocanon** 标志并忽略 **UseNocanon** 指令。
- 如果您为请求的 URL 定义上下文和 **ProxyPass** 指令，并且 **ProxyPass** 指令排除 **nocanon** 标志，代理会忽略 **UseNocanon** 指令。

注意

如果没有为请求的 URL 定义上下文，**mod_proxy_cluster** 会返回 **404** 错误。

C.3. MOD_ADVERTISE.SO 模块和指令

代理公告模块 **mod_advertise.so** 通过 UDP 多播消息广播代理服务器是否存在。服务器公告消息包含代理侦听希望加入负载平衡集群的节点的 IP 地址和端口号。

mod_advertise 模块必须与 **VirtualHost** 元素中的 **mod_manager** 模块一起定义。在以下示例中，**mod_advertise** 模块的标识符是 **advertise_module** ：

```
LoadModule advertise_module modules/mod_advertise.so
```

mod_advertise.so的可配置指令

mod_advertise 模块可使用以下指令进行配置：

ServerAdvertise

定义如何使用广告机制。
默认值为 **Off**。当设置为 **Off** 时，代理不会公告其位置。

当设置为 **On** 时，广告机制用于告知 worker 节点向这个代理发送状态信息。您还可以使用以下语法指定主机名和端口： **ServerAdvertise On http://*HOSTNAME*:*PORT*/**。只有在使用基于名称的虚拟主机或未定义虚拟主机时，才需要这样做。

AdvertiseGroup

定义要公告的多播地址。语法为 **AdvertiseGroup *ADDRESS* ：*PORT***，其中 ***ADDRESS*** 必须与 **AdvertiseGroupAddress** 对应，***PORT*** 必须与 worker 节点中的 **AdvertisePort** 对应。
如果您的 worker 节点基于 JBoss EAP，并且在启动时使用 **-u** 开关，则默认 **AdvertiseGroupAddress** 是通过 **-u** 开关传递的值。

默认值为 **224.0.1.105:23364**。如果没有指定端口，端口默认为 **23364**。

AdvertiseFrequency

多播消息之间的时间间隔（以秒为单位）公告 IP 地址和端口。默认值为 **10**。

AdvertiseSecurityKey

定义用于在 Apache HTTP 服务器中识别 **mod_proxy_cluster** 的字符串。默认情况下，不会设置这个指令，且不会发送任何信息。

AdvertiseManagerUrl

定义 worker 节点应该用来向代理服务器发送信息的 URL。默认情况下，不会设置这个指令，且不会发送任何信息。

AdvertiseBindAddress

定义用于发送多播消息的地址和端口。语法为 **AdvertiseBindAddress *ADDRESS*:*PORT***。这允许在多个 IP 地址的机器中指定地址。默认值为 **0.0.0.0:23364**。

C.4. MOD_CLUSTER_SLOTMEM.SO MODULE

mod_cluster_slotmem.so 模块是一个共享内存提供程序，用于创建和访问数据集以"slots"组织的共享内存段。

mod_cluster_slotmem 模块不需要任何配置指令。

C.5. 其他资源（或后续步骤）

- **mod_proxy_cluster**的 worker 节点配置参考

附录 D. MOD_JK的 WORKER.PROPERTIES 文件

如果要使用 **mod_jk** 连接器，您必须在 Apache HTTP **服务器主机上创建一个*JBCS_HOME*/httpd/conf/workers.properties** 文件来定义后端 worker 节点。worker 节点是 servlet 容器，您可以映射到 **mod_jk** 负载均衡器。**worker.properties** 文件指定 servlet 容器的位置，以及如何在这些 servlet 容器中平衡调用。

本附录描述了 **worker.properties** 文件的布局和内容。

根据您使用的 Apache HTTP 服务器分布，**worker.properties** 文件的位置会有所不同：

- 如果您使用 JBCS Apache HTTP 服务器，**worker.properties 文件位于 *JBCS_HOME*/httpd/conf.d** 目录中。
- 如果您使用 RHEL Apache HTTP 服务器，**worker.properties 文件位于 /etc/httpd/conf.d** 目录中。

注意

RHEL Apache HTTP 服务器只支持在 RHEL 9 上使用 **mod_jk**。

D.1. WORKERS.PROPERTIES 概述

worker.properties 文件包含全局 properties 部分和 worker properties 部分。

全局属性

本节包含适用于所有 worker 的指令。

worker 属性

本节包含适用于每个 worker 的指令。

每个节点都使用 worker 属性命名约定来定义。worker 名称只能包含小写字母、大写字母、数字和特定特殊字符(_、/)。

worker 属性的结构是 **worker。*WORKER_NAME.DIRECTIVE*。**

worker

所有 worker 属性的常量前缀。

WORKER_NAME

提供给 worker 的任意名称。例如： **node1**、**node_** 01、**Node_1**。

指令

需要的特定指令。

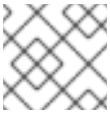
D.2. WORKERS.PROPERTIES 指令

worker.properties 文件指令被分成全局、强制、连接和负载平衡分类。

worker.properties 的全局指令

worker.list

指定 **mod_jk** 使用的 worker 名称列表。此列表中的 worker 可用于将请求映射到。



注意

不是由负载均衡器管理的单一节点配置必须设置为 **worker.list=*WORKER_NAME***。

worker.properties 的强制指令

type

指定 worker 类型，它决定适用于 worker 的指令。默认值为 **ajp13**，这是为 Web 服务器和 Apache HTTP 服务器之间的通信选择的首选 worker 类型。
其他值包括 **lb** 和 **status**。

有关 AJPv13 的详细信息，请参阅 Apache Tomcat Connector - AJP 协议参考 。

worker.properties的连接指令

host

worker 的主机名或 IP 地址。worker 节点必须支持 ajp13 协议堆栈。默认值为 **localhost**。
您可以通过在主机名或 IP 地址后附加端口号来指定 **port** 指令作为 host 指令的一部分。例如：**worker.node1.host=192.168.2.1:8009** 或 **worker.node1.host=node1.example.com:8009**。

端口

侦听定义的协议请求的远程服务器实例的端口号。默认值为 **8009**，这是 AJPv13 工作程序的默认监听端口。

ping_mode

指定连接在当前网络健康下探测到的条件。
该探测对 **CPing** 使用一个空的 AJPv13 数据包，并期望在一个特定的超时期间内，在返回中有一个 **CPong**。

您可以使用指令标志的组合来指定条件。标志不用逗号分开。例如，设置了正确的指令标志是 **worker.node1.ping_mode=CI**。

c （连接）

指定连接在连接到服务器后被探测到一次。您可以使用 **connect_timeout** 指令指定超时，否则会使用 **ping_timeout** 的值。

P (prepost)

指定连接在向服务器发送每个请求前被探测到。您可以使用 **prepost_timeout** 指令指定超时，否则会使用 **ping_timeout** 的值。

i (interval)

指定连接在常规内部维护周期内被探测到。您可以使用 **connection_ping_interval** 指令指定每个间隔之间的空闲时间，否则会使用 **ping_timeout** 的值。

A (all)

最常见的设置，它指定所有指令标志都已应用。有关 ***_timeout** 高级指令的详情，请查看 Apache Tomcat Connector - 参考指南。

ping_timeout

指定对一个 **CPing** 连接侦测的 **CPong** 回答的时间（请参阅 **ping_mode**）。默认值为 **10000** （毫秒）。

worker.properties 的负载均衡指令

lbfactor

指定单个 worker 的负载均衡因素，并只为负载均衡器的成员 worker 指定。
与集群中的其他 worker 相比，这个指令定义了分配给 worker 的 HTTP 请求负载的相对数量。

此指令应用的一个常见示例是，您希望将服务器与集群中的其他进程更多的处理能力区分开来。例如，如果您需要 worker 在负载超过其他 worker 三次，请指定 **worker。*WORKER_NAME*.lbfactor=3**。

balance_workers

指定负载均衡器必须管理的 worker 节点。该指令可以多次用于同一负载均衡器，它由 worker **.properties** 文件中指定的、以逗号分隔的 worker 名称列表组成。

sticky_session

指定是否将带有 SESSION ID 的 worker 请求路由到同一 worker。默认值为 **0** (false)。当设置为 **1** (true)时，启用负载均衡器持久性。
例如，如果您指定了 **worker.loadbalancer.sticky_session=0**，则每个请求在集群中的每个节点之间进行负载均衡。换句话说，同一会话的不同请求可以根据服务器负载访问不同的服务器。

如果您指定了 **worker.loadbalancer.sticky_session=1**，则每个会话都会保留（锁定）到一个服务器，直到会话终止为止，提供该服务器可用。

其他资源

- Apache Tomcat 连接器 - 参考指南 .

附录 E. MOD_PROXY_CLUSTER的 WORKER 节点配置参考

E.1. WORKER 节点配置

配置值在以下情况下发送到代理：

- 在服务器启动过程中
- 当通过广告机制检测到代理时
- 在重置代理配置时错误恢复过程中

表 E.1. Tomcat 的代理配置值

价值	Default（默认）	Description
stickySession	true	指定在可能的情况下，对给定会话的后续请求是否应路由到同一节点。
stickySessionRemove	false	如果负载均衡器无法将请求路由到它卡住的节点，则 Apache HTTP 服务器代理是否应该删除会话粘性。如果 **stickySession** 为 **false**，则此属性将被忽略。
stickySessionForce	true	指定当负载均衡器无法将请求路由到它被卡住的节点时，Apache HTTP 服务器代理是否应该返回错误。如果 **stickySession** 为 **false**，则此属性将被忽略。
workerTimeout	-1	指定 worker 可用于处理请求的秒数。当负载均衡器的所有 worker 都不可用时，**mod_proxy_cluster** 会在一个后重试(**workerTimeout/100**)来查找可用的 worker。值 **-1** 表示 Apache HTTP 服务器不会等待 worker 可用，并在没有 worker 可用时返回错误。
maxAttempts	1	指定 Apache HTTP 服务器代理在中止前尝试向 worker 发送给定请求的次数。最小值为 **1**： 在中止前尝试一次。
flushPackets	false	指定是否启用或禁用数据包清除。
flushWait	-1	指定清空数据包前等待的时间。值为 **-1** 表示等待永久。
ping	10	等待（以秒为单位）回答 ping 的时间。
smax		指定软最大闲置连接数。最大值由 Apache HTTP 服务器线程配置(**ThreadsPerChild** 或 **1**)决定。
ttl	60	指定 smax 阈值高于 **smax** 阈值的时间（以秒为单位）。

价值	Default（默认）	Description
nodeTimeout	-1	指定返回错误前，**mod_proxy_cluster** 在返回错误前等待后端服务器响应的时间（以秒为单位）。在转发请求前，**mod_proxy_cluster** 始终使用 **cping/cpong**。**mod_proxy_cluster** 使用的 **connectiontimeout** 值是 ping 值。
balancer	mycluster	指定负载均衡器的名称。
loadBalancingGroup		指定同一负载均衡组中的 **jvmRoutes** 之间的负载平衡。**负载均衡Group** 的概念相当于 mod_jk 域指令。

E.2. MOD_PROXY_CLUSTER的代理和代理发现配置属性

下表包含 **mod_proxy_cluster** 的代理和代理发现配置属性的属性和信息。

表 E.2. mod_proxy_cluster的代理发现配置属性

属性	属性	默认值
proxy-list	**proxyList**	
proxy-url	**proxyUrl**	
advertise	**advertise**	true
advertise-security-key	**advertiseSecurityKey**	
excluded-contexts	**excludedContexts**	
auto-enable-contexts	**autoEnableContexts**	true
stop-context-timeout	**stopContextTimeout**	10 秒（以秒为单位）
socket-timeout	**nodeTimeout**	20 秒（以毫秒为单位）

注意

如果没有定义 **nodeTimeout**，则使用 **ProxyTimeout** 指令 **Proxy**。如果没有定义 **ProxyTimeout**，则使用服务器超时（**Timeout**）（默认为 JBCS httpd.conf 中的120 秒）。**nodeTimeout**, **ProxyTimeout**, 和 **Timeout** 在套接字级别设置。

表 E.3. mod_proxy_cluster的代理配置属性

属性	属性	默认值
sticky-session	**stickySession**	true
sticky-session-remove	**stickySessionRemove**	false
sticky-session-force	**stickySessionForce**	true
node-timeout	**workerTimeout**	-1
max-attempts	**maxAttempts**	1
flush-packets	**flushPackets**	false
flush-wait	**flushWait**	-1
ping	**ping**	10 （秒）
smax	**smax**	-1 （使用默认值）
ttl	**ttl**	-1 （使用默认值）
domain	**loadBalancingGroup**	
load-balancing-group	**loadBalancingGroup**	

E.3. TOMCAT 的载入配置

当您要将 **mod_proxy_cluster** 与 Apache Tomcat 搭配使用时，您可以为加载指标配置以下附加属性。

表 E.4. Tomcat 的负载配置

属性	默认值	Description
loadMetricClass	**org.jboss.modcluster.load.metric.impl.BusyConnectorsLoadMetric**	实现 **org.jboss.load.metric.LoadMetric**的对象的类名称
loadMetricCapacity	1	通过 **loadMetricClass** 属性定义的负载指标的容量
loadHistory	9	在负载均衡因素计算中必须考虑的历史负载值数量
loadDecayFactor	2	历史负载值下降的因素

附录 F. 多处理模块(MPM)

Red Hat JBoss Core Services 包括各种多处理模块(MPM)。您可以使用这些 MPM 来自定义 Apache HTTP 服务器如何响应传入请求。

注意

MPM 是相互排斥的。您只能在任何特定时间启用和使用一个 MPM。

F.1. MPMS 概述

多处理模块(MPM)可用于 Red Hat Enterprise Linux (RHEL)和 Windows Server。在 RHEL 上，默认的 MPM 因操作系统版本而异。

用于 RHEL 的 MPM

prefork

prefork MPM 实现了一个非线程、预分叉的 web 服务器。 **prefork** MPM 使用单一控制进程，它会启动侦听和服务传入连接的子进程。单个进程处理特定的请求，以确保每个请求被隔离，不会影响任何其他请求。

注意

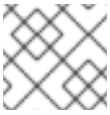
prefork MPM 是 RHEL 7 中默认的 MPM。

worker

worker MPM 实施混合多进程、多线程服务器。每个子进程创建固定数量的服务器线程，允许服务器处理大量系统资源的请求。

event

事件 MPM 基于 **worker** MPM。**事件** MPM 允许通过将某些处理工作委派给监听器线程来同时提供其他请求，从而释放 worker 线程来提供新请求。



注意

事件 MPM 是 RHEL 版本 8 和 9 的默认 MPM。

用于 Microsoft Windows 的 MPM

winnt

winnt MPM 是唯一可用于 Windows 系统的 MPM。 **winnt** MPM 使用单一控制进程，它启动另一个为传入请求创建线程的进程。

F.2. 切换 MPM

服务器根据 Apache HTTP 服务器主机上的 **00-mpm.conf** 文件中的 **LoadModule** 指令选择 MPM。您可以通过从 **00-mpm.conf** 文件中的 MPM 的 **LoadModule** 指令中删除注释字符(**#**)来选择特定的 MPM。

根据您使用的 Apache HTTP 服务器分发，00- **mpm.conf** 文件的位置会有所不同：

- 如果您使用 JBCS Apache HTTP 服务器，00- **mpm.conf 文件位于 *JBCS_HOME*/httpd/conf.modules.d** 目录中。
- 如果您使用 RHEL Apache HTTP 服务器，00- **mpm.conf 文件位于 /etc/httpd/conf.modules.d** 目录中。

根据您使用的操作系统版本，请考虑以下指南：

- 在 RHEL 版本 8 和 9 中，默认选择 **事件** MPM。例如：

```
# event MPM: A variant of the worker MPM with the goal of consuming
# threads only for connections with active processing
# See: http://httpd.apache.org/docs/2.4/mod/event.html
#
LoadModule mpm_event_module modules/mod_mpm_event.so
```

事件 MPM 是多线程的，旨在提供优化的性能。如果您使用 RHEL 版本 8 或 9，切换到另一个 MPM，如 **prefork** 可能会导致性能问题。

- 在 RHEL 7 中，默认选择 **prefork** MPM。例如：

```
# prefork MPM: Implements a non-threaded, pre-forking web server
# See: http://httpd.apache.org/docs/2.4/mod/prefork.html
LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
```

如果您使用 RHEL 7，请考虑切换到另一个 MPM，如 **worker** 或 **event**，以避免可能出现性能问题。

注意

为了说明目的，以下步骤描述了如何从 **prefork** MPM 切换到 **worker** MPM。

流程

1. 进入包含 **00-mpm.conf** 文件的目录：
   - 如果您使用 JBCS Apache HTTP 服务器，请转到 ***JBCS_HOME*/httpd/conf.modules.d** 目录。
   - 如果您使用 RHEL Apache HTTP 服务器，请转到 **/etc/httpd/conf.modules.d** 目录。
2. 编辑 **00-mpm.conf**，将注释(**#**)字符添加到 **prefork** MPM 的 **LoadModule** 指令中。
   例如：

```
# prefork MPM: Implements a non-threaded, pre-forking web server
# See: http://httpd.apache.org/docs/2.4/mod/prefork.html
#LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
```

3. 在同一个 **00-mpm.conf** 文件中，从您要切换到的 MPM 的 **LoadModule** 指令中删除注释(**#**)字符。这些行位于 **prefork** MPM 的下方。
   例如，要加载 **worker** MPM，请从 **worker** MPM 的 **LoadModule** 指令中删除注释(**#**)字符：

```
# worker MPM: Multi-Processing Module implementing a hybrid
# multi-threaded multi-process web server
# See: http://httpd.apache.org/docs/2.4/mod/worker.html
```

```
LoadModule mpm_worker_module modules/mod_mpm_worker.so
```

验证

- 要验证 MPM 是否已正确配置，请输入以下命令：

```
$ sbin/apachectl -V
```

前面的命令显示当前 MPM。

例如：

```
Server MPM:     worker
```

F.3. MPM 性能设置

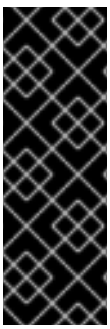
对于每种 MPM，您可以配置各种设置来优化 MPM 性能。

MPM 性能设置的类型
MPM 性能设置指定以下类型的条件：

- 启动时要创建的服务器进程的初始数量
- 最小和最大闲置线程或服务器进程数
- 处理请求的线程或服务器进程的最大数量
- 单个服务器进程可以处理的请求数
- 每个服务器进程创建的线程数（仅限**worker** 和 **事件** MPM）
- 对于在服务器生命周期内启动的最大服务器进程数上限（仅限**prefork** MPM）

MPM 性能设置的配置文件
在 JBCS 2.4.51 或更高版本中，您可以在 **mpm.conf** 文件中配置 MPM 性能设置。根据您使用的 Apache HTTP 服务器分发，**mpm.conf** 文件的位置会有所不同：

- 如果您使用 JBCS Apache HTTP 服务器，**mpm.conf** 文件位于 ***JBCS_HOME*/httpd/conf.d** 目录中。
- 如果您使用 RHEL Apache HTTP 服务器，**mpm.conf** 文件位于 **/etc/httpd/conf.d** 目录中。



重要

在 JBCS 2.4.37 或更早版本中，**conf.modules.d/00-mpm.conf** 文件包含 MPM 性能设置。从 JBCS 2.4.57 开始，**conf.d/mpm.conf** 文件包含这些设置。

如果您要从 JBCS 2.4.37 或更早版本升级，请确保为升级的 2.4.57 安装配置 **conf.d/mpm.conf** 文件，以匹配之前在 **conf.modules.d/00-mpm.conf** 中配置的任何自定义设置。否则，升级的 JBCS 2.4.57 安装自动使用 **conf.d/mpm.conf** 文件中的默认设置，这可能会导致意外的性能问题。

有关可用性能设置和相关默认值的更多信息，请参阅 Apache HTTP 服务器安装中的 **conf.d/mpm.conf** 文件。

其他资源

- Apache MPM Common Directives