



Red Hat Process Automation Manager 7.13

安装和配置 Red Hat Process Automation Manager

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

本文档论述了如何在红帽 JBoss EAP 上安装 Red Hat Process Automation Manager 组件，在红帽 JBoss EAP 集群环境中、JBoss Web 服务器和其他支持的环境中安装 Red Hat Process Automation Manager 组件。

目录

前言	6
使开源包含更多	7
部分 I. 规划 RED HAT PROCESS AUTOMATION 安装	8
第 1 章 RED HAT PROCESS AUTOMATION MANAGER 版本	9
第 2 章 RED HAT PROCESS AUTOMATION MANAGER 组件	10
第 3 章 RED HAT PROCESS AUTOMATION MANAGER 安装平台	11
3.1. RED HAT JBOSS EAP	11
3.2. RED HAT JBOSS WEB SERVER	11
3.3. RED HAT OPENSIFT CONTAINER PLATFORM	11
3.4. ORACLE WEBLOGIC SERVER	11
3.5. IBM WEBSHERE APPLICATION SERVER	12
第 4 章 RED HAT PROCESS AUTOMATION MANAGER 支持的软件仓库	13
4.1. GIT 存储库	13
4.2. APACHE MAVEN	13
第 5 章 RED HAT PROCESS AUTOMATION MANAGER 的安装环境选项	14
第 6 章 RED HAT PROCESS AUTOMATION MANAGER 角色和用户	15
第 7 章 RED HAT PROCESS AUTOMATION MANAGER 参考指南	16
第 8 章 与 RED HAT PROCESS AUTOMATION MANAGER 集成	17
8.1. RED HAT PROCESS AUTOMATION MANAGER SPRING BOOT 业务应用程序	17
8.2. 集成红帽 FUSE	17
8.3. 集成红帽单点登录	17
第 9 章 其他资源	18
部分 II. 在 RED HAT JBOSS EAP 7.4 上安装和配置 RED HAT PROCESS AUTOMATION MANAGER	19
第 10 章 关于 RED HAT PROCESS AUTOMATION MANAGER	20
第 11 章 RED HAT PROCESS AUTOMATION MANAGER 角色和用户	21
第 12 章 下载 RED HAT PROCESS AUTOMATION MANAGER 安装文件	22
第 13 章 使用 RED HAT PROCESS AUTOMATION MANAGER 安装程序	23
13.1. 在互动模式中使用安装程序	23
13.2. 在 CLI 模式中使用安装程序	26
13.3. 使用安装程序通过流程自动化管理器控制器配置 KIE 服务器	28
第 14 章 通过 ZIP 文件安装 RED HAT PROCESS AUTOMATION MANAGER	30
14.1. 从 ZIP 文件安装 BUSINESS CENTRAL	30
14.2. 从 ZIP 文件安装 KIE 服务器	31
14.3. 为 KIE 服务器配置 JDBC 数据源	32
14.4. 创建用户	35
14.5. 配置 KIE 服务器以连接至 BUSINESS CENTRAL	38
14.6. 线程效率	41
第 15 章 使用密钥存储保护密码	43

第 16 章 将 SSH 配置为使用 RSA	46
第 17 章 验证 RED HAT PROCESS AUTOMATION MANAGER 安装	47
第 18 章 运行 RED HAT PROCESS AUTOMATION MANAGER	49
第 19 章 安装并运行无头进程自动化管理器控制器	51
19.1. 安装无头进程自动化管理器控制器	51
19.2. 运行无头进程自动化管理器控制器	55
第 20 章 配置并运行独立 BUSINESS CENTRAL	57
第 21 章 RED HAT PROCESS AUTOMATION MANAGER 的 MAVEN 设置和软件仓库	59
21.1. 为 RED HAT PROCESS AUTOMATION MANAGER 添加 MAVEN 依赖项	59
21.2. 为 BUSINESS CENTRAL 和 KIE 服务器配置外部 MAVEN 存储库	63
21.3. 准备 MAVEN 镜像存储库以离线使用	65
第 22 章 从 GIT 存储库导入项目	69
第 23 章 自定义 BUSINESS CENTRAL 的品牌	70
23.1. 自定义 BUSINESS CENTRAL 登录页面	70
23.2. 自定义 BUSINESS CENTRAL 应用程序标头	70
第 24 章 集成 LDAP 和 SSL	72
附录 A. BUSINESS CENTRAL 系统属性	73
第 25 章 其他资源	82
第 26 章 后续步骤	83
部分 III. 在 RED HAT JBOSS EAP 集群环境中安装和配置 RED HAT PROCESS AUTOMATION MANAGER ..	84
第 27 章 RED HAT PROCESS AUTOMATION MANAGER 集群	85
第 28 章 在开发（授权）环境中的 RED HAT PROCESS AUTOMATION MANAGER 集群	86
28.1. 安装和配置红帽数据网格	87
28.2. 下载并配置 AMQ BROKER	88
28.3. 配置 NFS 版本 4 服务器	90
28.4. 下载并提取 RED HAT JBOSS EAP 7.4 和 RED HAT PROCESS AUTOMATION MANAGER	90
28.5. 在集群中配置和运行 BUSINESS CENTRAL	92
28.6. 测试您的高可用性(HA)内部基础架构	95
28.7. 验证 RED HAT PROCESS AUTOMATION MANAGER 集群	97
第 29 章 运行时环境中的 KIE 服务器集群	100
29.1. 下载并提取红帽 JBOSS EAP 7.4 和 KIE 服务器	100
29.2. 为 KIE 服务器配置并运行 RED HAT JBOSS EAP 7.4 集群	102
29.3. 使用无头进程自动化管理器控制器集群 KIE 服务器	106
29.4. 在失败的 KIE 服务器节点上为 RUNNING 状态配置作业故障切换	108
第 30 章 安装和配置智能路由器	112
30.1. 使用智能路由器负载均衡 KIE 服务器实例	112
30.2. 为 TLS 支持配置智能路由器	116
30.3. 为端点身份验证配置智能路由器	117
30.4. 配置智能路由器行为	118
30.5. 使用配置文件配置智能路由器设置	120
第 31 章 配置 QUARTZ 计时器服务	123

第 32 章 其他资源	126
部分 IV. 在 RED HAT JBOSS WEB SERVER 上安装和配置 RED HAT PROCESS AUTOMATION MANAGER	127
第 33 章 关于 RED HAT PROCESS AUTOMATION MANAGER	128
33.1. RED HAT PROCESS AUTOMATION MANAGER 组件	129
33.2. RED HAT PROCESS AUTOMATION MANAGER 角色和用户	130
第 34 章 下载 RED HAT PROCESS AUTOMATION MANAGER 安装文件	132
第 35 章 使用 RED HAT PROCESS AUTOMATION MANAGER 安装程序	134
35.1. 在互动模式中使用安装程序	134
35.2. 在 CLI 模式中使用安装程序	138
第 36 章 KIE SERVER ZIP 文件安装和配置	142
36.1. 从 ZIP 文件安装 KIE 服务器	142
36.2. 配置 JDBC WEB 服务器数据源	144
第 37 章 使用密钥存储保护密码	150
第 38 章 验证 KIE 服务器安装	153
第 39 章 下载并安装无头进程自动化管理器控制器	154
第 40 章 配置并运行独立 BUSINESS CENTRAL	157
第 41 章 RED HAT PROCESS AUTOMATION MANAGER 的 MAVEN 设置和软件仓库	159
41.1. 使用项目配置文件配置 MAVEN(POM.XML)	159
41.2. 修改 MAVEN 设置文件	159
41.3. 为 RED HAT PROCESS AUTOMATION MANAGER 添加 MAVEN 依赖项	160
41.4. 准备 MAVEN 镜像存储库以离线使用	164
第 42 章 从 GIT 存储库导入项目	168
第 43 章 集成 LDAP 和 SSL	169
附录 B. BUSINESS CENTRAL 系统属性	170
部分 V. 在 IBM WEBSHERE APPLICATION SERVER 上安装和配置 KIE 服务器	179
第 44 章 KIE 服务器	180
第 45 章 IBM WEBSHERE APPLICATION SERVER	181
第 46 章 安装和运行 IBM WEBSHERE APPLICATION SERVER	182
第 47 章 为 KIE 服务器配置 IBM WEBSHERE APPLICATION SERVER	184
47.1. 启用管理安全性	184
47.2. 在 IBM WEBSHERE APPLICATION SERVER 中配置 JDBC 数据源	185
47.3. 配置 JAVA 消息服务(JMS)	188
47.4. 在 IBM WEBSHERE APPLICATION SERVER 中设置系统属性	192
47.5. 停止并重启 IBM WEBSHERE APPLICATION SERVER	195
第 48 章 使用 IBM WEBSHERE 应用服务器安装并运行 KIE 服务器	197
48.1. 创建 KIE 服务器组和角色	198
48.2. 映射 KIE 服务器组和角色	199
48.3. 为 KIE 服务器配置类加载	200
48.4. 验证安装	201

第 49 章 使用 IBM WEBSPPHERE APPLICATION SERVER 安装并运行无头进程自动化管理器控制器	202
49.1. 为无外进程自动化管理器控制器配置类加载	204
49.2. 为无外进程自动化管理器控制器设置系统属性	204
49.3. 映射无头进程自动化管理器控制器组和角色	206
49.4. 验证安装	207
第 50 章 在 IBM WEBSPPHERE APPLICATION SERVER 中配置嵌入式进程引擎和决策引擎	209
第 51 章 使用密钥存储保护密码	214
第 52 章 后续步骤	216
部分 VI. 在 ORACLE WEBLOGIC 服务器中安装和配置 KIE 服务器	217
第 53 章 KIE 服务器	218
第 54 章 ORACLE WEBLOGIC SERVER	219
第 55 章 安装和运行 ORACLE WEBLOGIC SERVER	220
第 56 章 为 KIE 服务器配置 ORACLE WEBLOGIC SERVER	223
56.1. 配置 KIE 服务器组和用户	223
56.2. 在 ORACLE WEBLOGIC SERVER 中配置 JDBC 数据源	224
56.3. 配置 JAVA 消息服务(JMS)	228
56.4. 在 ORACLE WEBLOGIC SERVER 中设置系统属性	231
56.5. 停止和重启 ORACLE WEBLOGIC SERVER	233
第 57 章 使用 ORACLE WEBLOGIC SERVER 安装 KIE 服务器	235
57.1. 在 ORACLE WEBLOGIC SERVER 上验证 KIE 服务器安装	236
第 58 章 使用 ORACLE WEBLOGIC SERVER 安装并运行无头进程自动化管理器控制器	237
58.1. 为无外进程自动化管理器控制器设置系统属性	238
58.2. 验证安装	239
第 59 章 在 ORACLE WEBLOGIC SERVER 中配置嵌入式进程引擎和决策引擎	241
第 60 章 使用密钥存储保护密码	246
第 61 章 后续步骤	248
部分 VII. 修补和升级红帽流程自动化管理器	249
第 62 章 RED HAT PROCESS AUTOMATION MANAGER 版本	250
第 63 章 在红帽客户门户网站中启用通知	252
第 64 章 将补丁更新和次要发行升级到 RED HAT PROCESS AUTOMATION MANAGER	253
第 65 章 其他资源	259
附录 C. 版本信息	260
附录 D. 联系信息	261

前言

作为开发人员或系统管理员，您可以在 Red Hat JBoss EAP 集群环境、JBoss Web Server 和其他支持的环境中，在 Red Hat JBoss EAP 集群环境中安装 Red Hat Process Automation Manager 组件。

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

部分 I. 规划 RED HAT PROCESS AUTOMATION 安装

作为系统管理员，您有几个用于安装 Red Hat Process Automation 的选项。

第 1 章 RED HAT PROCESS AUTOMATION MANAGER 版本

Red Hat Process Automation Manager 版本使用数字 **Major.Minor.Patch** 格式指定，如 7.13.5。在本例中，主发行版本为 **7.x.x**，次要发行本是 **7.13.x**，补丁版本为 7.13.5。主发行版本通常需要数据迁移，而次版本升级和补丁更新通常通过 Red Hat Process Automation Manager 发行工件提供的更新工具进行管理。

以下是 Red Hat Process Automation Manager 的一般发行版本类型：

主发行版本迁移

Red Hat Process Automation Manager 的主要发行版本包括大量改进、安全更新、错误修复以及可能重新设计的功能和功能。当应用程序从一个主版本移动到另一个主版本（如从 Red Hat JBoss BPM Suite 6.4.x）迁移到 Red Hat Process Automation Manager 7.0 时，通常需要数据迁移。自动化迁移工具通常附带新的 Red Hat Process Automation Manager 主版本，以便促进迁移，但某些数据和配置可能需要一些手动工作。支持的迁移路径在产品公告和文档中指定。

有关迁移说明，[请参阅从 Red Hat JBoss BPM Suite 6.4 迁移到 Red Hat Process Automation Manager 7.0。](#)

次发行版本升级

Red Hat Process Automation Manager 的次发行版本包括增强、安全更新和程序错误修复。当应用程序从一个次版本移到另一个次版本时，可能需要数据迁移，如从 Red Hat Process Automation Manager 7.5.x 升级到 7.6。Red Hat Process Automation Manager 的补丁更新和新的次要版本中通常会提供自动更新，以帮助更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE 服务器和无头进程自动化管理器控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）都会作为新的工件发布，且您必须重新安装它们以应用更新。

升级到新的次版本之前，请将最新的补丁更新应用到 Red Hat Process Automation Manager 的最新版，以确保次要发行本升级成功。

补丁更新

Red Hat Process Automation Manager 的补丁更新包括最新的安全更新和程序错误修复。计划的补丁更新包含之前为该产品次要版本发布的所有补丁更新，因此您无需逐步应用各个补丁更新，以应用最新的更新。例如，您可以将 Red Hat Process Automation Manager 7.5.0 或 7.5.1 更新至 Red Hat Process Automation Manager 7.5.2。但是，为了获得最佳 Red Hat Process Automation Manager 性能，在产品可用时应用产品更新。有时，红帽可能会在现有产品的正常更新周期外发布非计划补丁更新。这可能包括安全性或其他由 Red Hat 全球支持服务(GSS)提供的更新，以便修复特定问题，且可能不是累积更新。

Red Hat Process Automation Manager 的补丁更新和新的次要版本中通常会提供自动更新，以帮助更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE 服务器和无头进程自动化管理器控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）都会作为新的工件发布，且您必须重新安装它们以应用更新。

为确保在版本间进行最佳过渡，并使用最新的改进和修补程序保持 Red Hat Process Automation Manager，在红帽客户门户网站中可用，对 Red Hat Process Automation Manager 应用新的产品版本和更新。请考虑在红帽客户门户网站中启用产品通知。

第 2 章 RED HAT PROCESS AUTOMATION MANAGER 组件

该产品由 Business Central 和 KIE 服务器组成。

- Business Central 是您创建和管理业务规则的图形用户界面。您可以在 Red Hat JBoss EAP 实例或 Red Hat OpenShift Container Platform(OpenShift)上安装 Business Central。Business Central 也作为独立 JAR 文件提供。您可以使用 Business Central 独立 JAR 文件运行 Business Central，而无需将其部署到应用服务器。
- KIE 服务器是执行规则和其他工件的服务器。它用于实例化和执行规则并解决规划问题。您可以在红帽 JBoss EAP 实例中安装 KIE 服务器（在红帽 JBoss EAP 集群中），在 OpenShift 上、Oracle WebLogic 服务器实例、IBM WebSphere Application Server 实例或作为 Spring Boot 应用程序的一部分。
您可以将 KIE 服务器配置为在受管或非受管模式下运行。如果 KIE 服务器为非受管，您必须手动创建和维护 KIE 容器（部署单元）。KIE 容器是项目的特定版本。如果管理 KIE 服务器，则“流程自动化管理器”控制器管理 KIE 服务器配置，并与流程自动化管理器控制器交互以创建和维护 KIE 容器。

流程自动化管理器控制器与 Business Central 集成。如果您在红帽 JBoss EAP 上安装 Business Central，请使用执行服务器页面来创建和维护 KIE 容器。但是，如果您不安装 Business Central，则可安装无头流程自动化管理器控制器，并使用 REST API 或 KIE 服务器 Java 客户端 API 与它交互。

- 红帽构建的 OptaPlanner 集成到 Business Central 和 KIE Server 中。它是一种轻量级、可嵌入的规划引擎，可优化规划问题。红帽构建的 OptaPlanner 有助于 Java 编程人员有效地解决规划问题，并合并优化的 heuristics 和 metaheuristics，并通过有效的分数计算来合并优化 heuristics 和 metaheuristics。

第 3 章 RED HAT PROCESS AUTOMATION MANAGER 安装平台

每个 Red Hat Process Automation Manager 的每个发行版本都在各种操作系统、JVM、Web 浏览器和数据库组合中认证。根据您的订阅协议，红帽为物理和虚拟环境中的受支持和经过测试的配置提供生产和开发支持。有关支持的配置和版本号，请参见以下页面：

- [Red Hat Process Automation Manager 7 支持的配置](#)
- [Red Hat Process Automation Manager 7 组件详情](#)

Red Hat Process Automation Manager 7.13 可用于以下应用程序平台：

- Red Hat JBoss EAP
- Red Hat JBoss Web Server
- Red Hat OpenShift Container Platform
- Oracle WebLogic Server
- IBM WebSphere Application Server

3.1. RED HAT JBOSS EAP

Red Hat JBoss Enterprise Application Platform(Red Hat JBoss EAP)是 Java Enterprise Edition 7(Java EE 7)的已认证实施，也是 Web 配置文件规格。红帽 JBoss EAP 为高可用性、集群、消息传递和分布式缓存等功能提供预配置选项。它还允许用户使用 Red Hat JBoss EAP 提供的各种 API 和服务编写、部署和运行应用程序。

您可以在单个红帽 JBoss EAP 实例中安装 Business Central 和 KIE 服务器。但是，在生产环境中，您应该在单独的实例中安装它们。

3.2. RED HAT JBOSS WEB SERVER

红帽 JBoss Web 服务器是一个企业级 Web 服务器，专为中型和大型应用程序而设计，基于 Tomcat。红帽 JBoss Web 服务器为组织提供了用于 Java 服务器页面(JSP)和 Java Servlet 技术、PHP 和 CGI 的单一部署平台。

您可以在 Red Hat JBoss Web Server 上安装 KIE 服务器和无头进程自动化管理器控制器。

3.3. RED HAT OPENSIFT CONTAINER PLATFORM

Red Hat OpenShift Container Platform(OpenShift)使用 Kubernetes，并提供一个 API 来管理这些服务。OpenShift 允许您创建和管理容器。

在这个解决方案中，Red Hat Process Automation Manager 的组件被部署为独立的 OpenShift pod。您可以单独扩展每个 pod，并尽可能为特定组件提供几个或数量个容器。您可以使用标准的 OpenShift 方法来管理 pod，并平衡负载。

您可以在 OpenShift 上安装 Business Central 和 KIE 服务器。

3.4. ORACLE WEBLOGIC SERVER

Oracle WebLogic Server 是 Java EE 应用服务器，提供标准的 API 集，用于创建分布式 Java 应用程序，可访问各种服务，如数据库、消息服务和与外部企业系统的连接。用户使用 Web 浏览器客户端或 Java 客户端访问这些应用程序。

您可以在 Oracle WebLogic Server 实例中安装 KIE 服务器，然后使用无头进程自动化管理器控制器和 REST API 或 KIE 服务器 Java 客户端 API 与 KIE 服务器交互。您还可以使用独立 Business Central 与 KIE 服务器交互。

3.5. IBM WEBSPHERE APPLICATION SERVER

IBM WebSphere Application Server 是一个灵活、安全的 Web 应用程序服务器，托管基于 Java 的 Web 应用程序，并提供 Java EE 认证的运行时环境。IBM WebSphere 9.0 支持 Java SE 8，且从版本 8.5.5.6 开始完全兼容 Java EE 7。

您可以在 IBM WebSphere Application Server 实例中安装 KIE 服务器，然后使用无外进程自动化管理器控制器与 KIE 服务器交互。您还可以使用独立 Business Central 与 KIE 服务器交互。

第 4 章 RED HAT PROCESS AUTOMATION MANAGER 支持的软件仓库

您可以将 Git 存储库和 Apache Maven 与 Red Hat Process Automation Manager 搭配使用。

4.1. GIT 存储库

Git 存储库在 Business Central 内部使用，以存储编写环境中创建的所有流程、规则和其他工件。Git 是一种分布式版本控制系统。它以提交对象的形式实施修订版本。当您将更改提交到存储库时，会创建一个新的提交对象。当您在 Business Central 中创建项目时，它会被添加到连接到 Business Central 的 Git 存储库中。

如果您在其他 Git 存储库中拥有项目，您可以将其导入到 Business Central 空间中。您可以使用 Git hook 将内部 Git 存储库与外部存储库同步。

4.2. APACHE MAVEN

[Apache Maven](#) 是一个分布式构建自动化工具，用于构建和管理软件项目。您可以使用 Maven 构建、发布和部署 Red Hat Process Automation Manager 项目。Maven 提供以下优点：

- 构建过程比较简单，统一的构建系统是在项目之间实施的。
- 在编译时，项目所需的所有 JAR 文件都可用。
- 配置了适当的项目结构。
- 依赖项和版本是很好的管理。
- 不需要额外的构建处理，因为 Maven 构建输出到多个预定义类型，如 JAR 和 WAR。

Maven 使用存储库来存储 Java 库、插件和其他构建工件。这些仓库可以是本地或远程的。Red Hat Process Automation Manager 维护本地和远程 maven 存储库，您可以添加到项目中以访问规则、流程、事件和其他项目依赖项。在构建项目和 archetypes 时，Maven 从本地或远程存储库动态检索 Java 库和 Maven 插件。这样做可促进跨项目共享和重复利用依赖项。

第 5 章 RED HAT PROCESS AUTOMATION MANAGER 的安装环境选项

使用 Red Hat Process Automation Manager，您可以设置一个开发环境以开发业务应用程序，一个运行这些应用程序的运行时环境来支持决策。

- **开发环境**：通常由一个 Business Central 安装和一个 KIE 服务器安装组成。您可以使用 Business Central 设计决策和其他工件，您可以使用 KIE 服务器执行和测试您所创建的工件。
- **运行时环境**：通过或不使用 Business Central 实现一个或多个 KIE 服务器实例。Business Central 具有一个嵌入式流程自动化管理器控制器。如果您安装 Business Central，使用 **Menu → Deploy → Execution servers** 页面来创建和维护容器。如果要在没有 Business Central 的情况下自动执行 KIE 服务器管理，您可以使用无头流程自动化管理器控制器。

您还可以集群开发和运行时环境。集群开发或运行时环境由两个或多个服务器的统一组或集群组成。集群红帽流程自动化管理器开发环境的主要优点是高可用性和增强的协作，而集群红帽流程自动化管理器运行时环境的主要优点是高可用性和负载平衡。当单个服务器出现故障时，高可用性降低了数据丢失的可能性。当服务器出现故障时，另一个服务器通过提供故障服务器上的数据副本来填补差距。当故障服务器再次上线时，它会在集群中恢复它。



注意

目前，Red Hat JBoss EAP 7.4 和 Red Hat OpenShift Container Platform 支持运行时环境的集群。

第 6 章 RED HAT PROCESS AUTOMATION MANAGER 角色和用户

要访问 Business Central 或 KIE 服务器，您必须在服务器启动前创建用户并为其分配适当的角色。安装 Business Central 或 KIE 服务器时，您可以创建用户和角色。

如果 Business Central 和 KIE 服务器都在单个实例上运行，则为 Business Central 进行身份验证的用户也可以访问 KIE 服务器。

然而，如果 Business Central 和 KIE 服务器在不同的实例上运行，则必须单独验证用于 Business Central 的用户才能访问 KIE 服务器。例如，如果在 Business Central 上进行身份验证但未在 KIE Server 进行身份验证的用户，如果在 Business Central 中查看或管理进程定义，则 401 错误会在日志文件中记录，而**无效凭证来从远程服务器加载数据**。请联系您的系统管理员。信息会出现在 Business Central 中。

本节论述了 Red Hat Process Automation Manager 用户角色。



注意

Business Central 保留了管理员、分析师、开发人员、经理、**process-admin**、**user** 和 **rest-all** 角色。**kie-server** 角色保留用于 KIE 服务器。因此，根据 Business Central、KIE 服务器还是安装这两者，可用角色可能会有所不同。

- **管理**：具有**管理员角色**的用户是 Business Central 管理员。他们可以管理用户、创建、克隆和管理存储库。他们可以完全访问，以便对应用程序进行必要的更改。具有 **admin** 角色的用户可以访问 Red Hat Process Automation Manager 中所有区域。
- **分析师**：具有 **analyst** 角色的用户可以访问所有高级功能。它们可以建模和执行其项目。但是，这些用户无法在 **Design → Projects** 视图中的空格中添加投稿者。对用于管理员的 **Deploy → Execution Servers** 视图的访问，它不适用于具有 **analyst** 角色的用户。但是，当他们访问 Library 视角时，这两个用户可使用 **Deploy** 按钮。
- **开发人员**：具有 **开发人员** 角色的用户可以访问几乎所有的功能，并可管理规则、模型、流程流、表单和仪表盘。它们可以管理资产存储库，他们可以创建、构建和部署项目。只有某些管理功能（如创建和克隆新存储库）才会从具有 **developer** 角色的用户中隐藏。
- **管理器**：具有 **manager** 角色的用户可以查看报告。这些用户通常对有关业务流程及其性能、业务指标和其他业务相关的报告的统计信息感兴趣。具有此角色的用户只能访问进程和任务报告。
- **process-admin**：具有 **process-admin** 角色的用户是业务流程管理员。他们可以完全访问业务流程、业务任务和执行错误。这些用户也可以查看业务报告，并有权访问 Task Inbox 列表。
- **用户**：具有 **用户** 角色的用户可以处理 Task Inbox 列表，其中包含目前正在运行进程的业务任务。具有此角色的用户可以查看进程和任务报告并管理进程。
- **REST-all**：具有 **restall** 角色的用户可以访问 Business Central REST 功能。
- **kie-server**：具有 **kie-server** 角色的用户可以访问 KIE 服务器 REST 功能。该角色是必需的，用户必须有权访问 Business Central 中的 **管理** 和 **跟踪** 视图。

第 7 章 RED HAT PROCESS AUTOMATION MANAGER 参考指南

Red Hat Process Automation Manager 提供了可用作入门应用程序的参考实现。它们包含在红帽客户门户网站中的 [Red Hat Process Automation Manager 7.13.5 Reference Implementations](#) 下载中。

员工级别参考实施

员工名单的参考实施允许您创建一个应用程序，为员工分配组织内不同职位的应用程序。例如，您可以使用应用程序在 nurses、guard duty 跨多个位置转移，或在 worker 之间的装配行转移。

载体路由规划参考实施

借助车辆路由规划参考实施，您可以创建一个应用程序，通过现实地图、路路和车辆来为位置提供良好位置、各容量来解决工具路由规划问题。如需更多信息，请参阅资源间路由 ZIP 文件中的 README 文件，包括在参考实施下载中。

院校时间参考实施

课时参考实施允许您在 Spring Boot 上构建 REST 应用程序，该应用程序与房间和时间插槽相关联，并通过考虑学员和指导限制来避免冲突。

高可用性事件驱动决策参考实施

借助高可用性事件驱动的决策参考实现，您可以在 OpenShift 环境中部署需要有状态处理（包括使用复杂事件处理功能的规则）的 Drools 引擎代码。这样做可让决策引擎处理具有高可用性的复杂事件系列。

第 8 章 与 RED HAT PROCESS AUTOMATION MANAGER 集成

您可以将 Red Hat Process Automation Manager 与其他产品和组件集成，如 Spring Boot、Red Hat Fuse 和 Red Hat Single Sign-On。

8.1. RED HAT PROCESS AUTOMATION MANAGER SPRING BOOT 业务应用程序

Spring Framework 是一个 Java 平台，可为开发 Java 应用程序提供全面的基础架构支持。Spring Boot 是基于 Spring Boot 启动器的轻量级框架。Spring Boot starters 是 `pom.xml` 文件，其中包含一组可在 Spring Boot 项目中包括的依赖项描述符。作为 Red Hat Runtimes 的一部分，红帽为 Red Hat Process Automation Manager Spring Boot 项目提供支持和维护。

Red Hat Process Automation Manager Spring Boot 业务应用程序是灵活的 UI 的逻辑分组，可提供特定业务功能。商业应用程序基于 Spring Boot 启动器。它们通常单独部署，可以单独使用。完整的业务应用程序可让域实现特定的业务目标，例如订单管理或调整管理。在创建和配置业务应用程序后，您可以通过 OpenShift 将其部署到现有服务或云。

8.2. 集成红帽 FUSE

Fuse 是一个分布式、云原生的集成平台，作为敏捷集成解决方案的一部分。其分布式方法使团队能够根据需要部署集成服务。

您可以在 Apache Karaf 容器平台或 Red Hat JBoss Enterprise Application Platform 上安装 Red Hat Fuse，然后在该容器中安装和配置 Red Hat Process Automation Manager。

8.3. 集成红帽单点登录

Red Hat Single Sign-On(RH-SSO)是一个单点登录解决方案，您可以使用 REST Web 服务和 Git 访问来保护浏览器应用程序。

当您为 Red Hat Process Automation Manager 与 RH-SSO 集成时，您可以为 Red Hat Process Automation Manager 创建 SSO 和身份管理(IDM)环境。RH-SSO 的会话管理功能可让您在互联网上针对不同的 Red Hat Process Automation Manager 环境使用单一身份验证。

Red Hat Process Automation Manager 在 Red Hat JBoss EAP 7.4 上支持 RH-SSO 7.5。

第 9 章 其他资源

安装

- [在 Red Hat JBoss EAP 7.4 上安装和配置 Red Hat Process Automation Manager](#)
- [在 Red Hat JBoss EAP 集群环境中安装和配置 Red Hat Process Automation Manager](#)
- [在 Red Hat JBoss Web Server 上安装和配置 Red Hat Process Automation Manager](#)
- [在 IBM WebSphere Application Server 上安装和配置 KIE 服务器](#)
- [在 Oracle WebLogic 服务器中安装和配置 KIE 服务器](#)

集成

- [使用 Spring Boot 创建 Red Hat Process Automation Manager 业务应用程序](#)
- [将 Red Hat Fuse 与红帽流程自动化管理器集成](#)
- [将 Red Hat Process Automation Manager 与红帽单点登录集成](#)

红帽构建的 OptaPlanner

- [使用红帽流程自动化管理器开发 Solvers](#)

OpenShift

- [使用 Operator 在 Red Hat OpenShift Container Platform 4 上部署 Red Hat Process Automation Manager 环境](#)
- [使用模板在 Red Hat OpenShift Container Platform 3 上部署 Red Hat Process Automation Manager 环境](#)

部分 II. 在 RED HAT JBOSS EAP 7.4 上安装和配置 RED HAT PROCESS AUTOMATION MANAGER

本文档论述了如何在 Red Hat JBoss Enterprise Application Platform 7.4 实例中安装 Red Hat Process Automation Manager。

先决条件

- 您已查看 [规划 Red Hat Process Automation Manager 安装中的信息](#)。
- 已安装 Red Hat JBoss Enterprise Application Platform 7.4 的最新补丁版本。有关安装红帽 JBoss EAP 的详情，请查看 [Red Hat JBoss Enterprise Application Platform 安装指南](#)。

第 10 章 关于 RED HAT PROCESS AUTOMATION MANAGER

Red Hat Process Automation Manager 是用于创建业务自动化应用程序和微服务的红帽中间件平台。它使企业业务和 IT 用户能够记录、模拟、管理、自动化和监控业务流程和策略。它旨在使商业和 IT 用户能够更有效地合作，从而轻松、快速地更改业务应用程序。

该产品由 Business Central 和 KIE 服务器组成。

- KIE 服务器是执行规则和其他工件的服务器。它用于实例化和执行规则并解决规划问题。KIE 服务器为业务资产提供运行时环境，并访问存储在资产存储库中的数据（知识库）。
- Business Central 是用来创建和管理 KIE 服务器执行的业务规则的图形用户界面。它可让您执行以下任务：
 - 创建、管理和编辑您的规则、流程和相关资产。
 - 管理连接的 KIE 服务器实例及其 KIE 容器（部署单元）。
 - 针对连接到 Business Central 的 KIE 服务器实例中的流程和任务执行运行时操作。
Business Central 也作为独立 JAR 文件提供。您可以使用 Business Central 独立 JAR 文件来运行 Business Central，而无需将其部署到应用服务器。

Red Hat JBoss Enterprise Application Platform (Red Hat JBoss EAP) 7.4 是 Java Enterprise Edition 8 (Java EE 8) 的完整和 Web 配置文件规格的已认证实施。红帽 JBoss EAP 为高可用性、集群、消息传递和分布式缓存等功能提供预配置选项。它还允许用户使用 Red Hat JBoss EAP 提供的各种 API 和服务编写、部署和运行应用程序。

本文档解释了如何在 Red Hat JBoss EAP 7.4 服务器实例中安装 Red Hat Process Automation Manager。

有关如何在其他环境中安装 Red Hat Process Automation Manager 的说明，请参阅以下文档：

- [在 IBM WebSphere Application Server 上安装和配置 KIE 服务器](#)
- [在 Oracle WebLogic 服务器中安装和配置 KIE 服务器](#)
- [使用 Operator 在 Red Hat OpenShift Container Platform 4 上部署 Red Hat Process Automation Manager 环境](#)
- [使用模板在 Red Hat OpenShift Container Platform 3 上部署 Red Hat Process Automation Manager 环境](#)

有关支持的组件的详情，请查看以下文档：

- [Red Hat Process Automation Manager 和 Maven 库版本之间的映射是什么？](#)
- [Red Hat Process Automation Manager 7 支持的配置](#)

第 11 章 RED HAT PROCESS AUTOMATION MANAGER 角色和用户

要访问 Business Central 或 KIE 服务器，您必须在服务器启动前创建用户并为其分配适当的角色。安装 Business Central 或 KIE 服务器时，您可以创建用户和角色。

如果 Business Central 和 KIE 服务器都在单个实例上运行，则为 Business Central 进行身份验证的用户也可以访问 KIE 服务器。

然而，如果 Business Central 和 KIE 服务器在不同的实例上运行，则必须单独验证用于 Business Central 的用户才能访问 KIE 服务器。例如，如果在 Business Central 上进行身份验证但未在 KIE Server 进行身份验证的用户，如果在 Business Central 中查看或管理进程定义，则 401 错误会在日志文件中记录，而**无效凭证来从远程服务器加载数据**。请联系您的系统管理员。信息会出现在 Business Central 中。

本节论述了 Red Hat Process Automation Manager 用户角色。



注意

Business Central 保留了管理员、分析师、开发人员、经理、**process-admin**、**user** 和 **rest-all** 角色。**kie-server** 角色保留用于 KIE 服务器。因此，根据 Business Central、KIE 服务器还是安装这两者，可用角色可能会有所不同。

- **管理**：具有**管理员角色**的用户是 Business Central 管理员。他们可以管理用户、创建、克隆和管理存储库。他们可以完全访问，以便对应用程序进行必要的更改。具有 **admin** 角色的用户可以访问 Red Hat Process Automation Manager 中所有区域。
- **分析师**：具有 **analyst** 角色的用户可以访问所有高级功能。它们可以建模和执行其项目。但是，这些用户无法在 **Design → Projects** 视图中的空格中添加投稿者。对用于管理员的 **Deploy → Execution Servers** 视图的访问，它不适用于具有 **analyst** 角色的用户。但是，当他们访问 Library 视角时，这两个用户可使用 **Deploy** 按钮。
- **开发人员**：具有 **开发人员** 角色的用户可以访问几乎所有的功能，并可管理规则、模型、流程流、表单和仪表盘。它们可以管理资产存储库，他们可以创建、构建和部署项目。只有某些管理功能（如创建和克隆新存储库）才会从具有 **developer** 角色的用户中隐藏。
- **管理器**：具有 **manager** 角色的用户可以查看报告。这些用户通常对有关业务流程及其性能、业务指标和其他业务相关的报告的统计信息感兴趣。具有此角色的用户只能访问进程和任务报告。
- **process-admin**：具有 **process-admin** 角色的用户是业务流程管理员。他们可以完全访问业务流程、业务任务和执行错误。这些用户也可以查看业务报告，并有权访问 Task Inbox 列表。
- **用户**：具有 **用户** 角色的用户可以处理 Task Inbox 列表，其中包含目前正在运行进程的业务任务。具有此角色的用户可以查看进程和任务报告并管理进程。
- **REST-all**：具有 **restall** 角色的用户可以访问 Business Central REST 功能。
- **kie-server**：具有 **kie-server** 角色的用户可以访问 KIE 服务器 REST 功能。该角色是必需的，用户必须有权访问 Business Central 中的 **管理** 和 **跟踪** 视图。

第 12 章 下载 RED HAT PROCESS AUTOMATION MANAGER 安装文件

您可以使用安装程序 JAR 文件或可部署 ZIP 文件来安装 Red Hat Process Automation Manager。您可以在互动或命令行界面(CLI)模式下运行安装程序。或者，您可以提取并配置 Business Central 和 KIE 服务器可部署 ZIP 文件。如果您想在不将其部署到应用程序服务器的情况下运行 Business Central，请下载 Business Central Standalone JAR 文件。

下载符合您环境和安装要求的 Red Hat Process Automation Manager 发行版本。

流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.13.5
2. 根据您的首选安装方法下载以下产品分发之一：



注意

您只需要下载这些发行版之一。

- 如果要使用安装程序在 Red Hat JBoss EAP 7.4 上安装 Red Hat Process Automation Manager，请下载 **Red Hat Process Automation Manager 7.13.5 Installer(rhpam-installer-7.13.5.jar)**。安装程序图形用户界面帮助您完成安装过程。
- 如果要使用可部署资源 ZIP 文件在 Red Hat JBoss EAP 7.4 上安装 Red Hat Process Automation Manager，请下载以下文件：
 - 用于所有支持的 EE8 容器的 Red Hat Process Automation Manager 7.13.5 KIE Server (rhpam-7.13.5-kie-server-ee8.zip)
 - Red Hat Process Automation Manager 7.13.5 Business Central Deployable for EAP 7 (rhpam-7.13.5-business-central-eap7-deployable.zip)
 - Red Hat Process Automation Manager 7.13.5 Add Ons(rhpam-7.13.5-add-ons.zip)
- 要运行 Business Central，无需将它部署到应用服务器，请下载 **Red Hat Process Automation Manager 7.13.5 Business Central Standalone (rhpam-7.13.5-business-central-standalone.jar)**。

第 13 章 使用 RED HAT PROCESS AUTOMATION MANAGER 安装程序

这部分论述了如何使用安装程序 JAR 文件安装 Business Central 和 KIE 服务器。JAR 文件是一个可执行文件，可在现有 Red Hat JBoss EAP 7.4 服务器中安装 Red Hat Process Automation Manager。您可以在互动或命令行界面(CLI)模式下运行安装程序。



注意

Red Hat Process Automation Manager JAR 文件安装程序 *不支持* yum 或 RPM Package Manager 安装的 Red Hat JBoss EAP 发行版本。如果要在此类 Red Hat JBoss EAP 安装中安装 Red Hat Process Automation Manager，请下载 **Red Hat Process Automation Manager 7.13 Deployable for Red Hat JBoss EAP 7.4** 文件，并按照 [第 14 章 通过 ZIP 文件安装 Red Hat Process Automation Manager](#) 中的步骤操作。



注意

因为 IBM JDK 无法使用在其他 JDK 上生成的密钥存储，所以您无法将 Red Hat Process Automation Manager 安装到在 IBM JDK 上运行的现有红帽 JBoss EAP 安装中，且密钥存储在另一个 JDK 上生成的密钥存储。

后续步骤：

按照以下部分中的说明操作：

- [第 13.1 节 “在互动模式中使用安装程序”](#)
- [第 13.2 节 “在 CLI 模式中使用安装程序”](#)
- [第 13.3 节 “使用安装程序通过流程自动化管理器控制器配置 KIE 服务器”](#)

13.1. 在互动模式中使用安装程序

Red Hat Process Automation Manager 的安装程序是一个可执行 JAR 文件。您可以使用它在现有 Red Hat JBoss EAP 7.4 服务器安装中安装 Red Hat Process Automation Manager。



注意

为安全起见，您应该以非 root 用户身份运行安装程序。

先决条件

- **Red Hat Process Automation Manager 7.13.5 安装程序** 已下载。具体说明请查看 [第 12 章 下载 Red Hat Process Automation Manager 安装文件](#)。
- 已安装受支持的 JDK。有关支持的 JDK 列表，请参阅 [Red Hat Process Automation Manager 7 支持的配置](#)。
- 已备份的红帽 JBoss EAP 7.4 服务器安装可用。
- 需要足够的用户权限以完成安装。
- JAR 二进制文件包含在 `$PATH` 环境变量中。在 Red Hat Enterprise Linux 上，它包含在 `java-$JAVA_VERSION-openjdk-devel` 软件包中。



注意

Red Hat Process Automation Manager 旨在使用 UTF-8 编码。如果底层 JVM 使用了不同的编码系统，则可能发生意外错误。为确保 JVM 使用 UTF-8，请使用 `"-Dfile.encoding=UTF-8"` 系统属性。有关系统属性列表，请查看 [附录 A, Business Central 系统属性](#)。

流程

1. 在终端窗口中，导航到您下载安装程序 JAR 文件并输入以下命令的目录：

```
java -jar rhpam-installer-7.13.5.jar
```



注意

在 Windows 上运行安装程序时，可能会提示您在安装过程中提供管理员凭证。要防止这个要求，请在安装命令中添加 `izpack.mode=privileged` 选项：

```
java -Dizpack.mode=privileged -jar  
rhpam-installer-7.13.5.jar
```

另外，当在 32 位 Java 虚拟机上运行安装程序时，您可能会遇到内存限制。要防止这个问题，请运行以下命令：

```
java -XX:MaxHeapSize=4g -jar  
rhpam-installer-7.13.5.jar
```

图形安装程序显示启动画面和许可证协议页面。

2. 点击 **我接受此许可协议的条款**，然后单击下一步。
3. 指定您要安装 Red Hat Process Automation Manager 的 Red Hat JBoss EAP 7.4 服务器主页，然后点 **Next**。
4. 选择您要安装的组件，然后点 **Next**。



注意

您可以在同一服务器上安装 Business Central 和 KIE 服务器。然而，您应该在生产环境中在不同的服务器上安装 Business Central 和 KIE 服务器。为此，请运行安装程序两次。

5. 创建一个用户，然后单击下一步。默认情况下，如果您在同一容器中同时安装 Business Central 和 KIE 服务器，则会为新用户提供 **管理员**、**kie-server** 和 **rest-all** 角色。如果您只安装 KIE 服务器，则向用户提供 **kie-server** 角色。若要选择其他角色，请取消选择 **admin**。有关角色的详情，请参考 [第 11 章 Red Hat Process Automation Manager 角色和用户](#)。



注意

确保指定的用户名与现有用户、角色或组不同。例如，不要创建用户名为 **admin** 的用户。

密码必须至少包含八个字符，且必须至少包含一个数字和一个非字母数字字符，但不包括和 (ampersand)。

记录用户名和密码。您将需要他们访问 Business Central 和 KIE 服务器。

6. 在 **Installation Overview** 页面中，查看您要安装的组件，然后点 **Next** 开始安装。
7. 安装完成后，单击 **下一步**。
8. 如果安装了 KIE 服务器，则“**配置运行时**”步骤在“**组件安装**”下显示。在 **Configure Runtime Environment** 页面中，选择执行默认安装或执行高级配置。
如果您选择 **执行高级配置**，可以选择配置数据库设置或自定义某些 KIE 服务器选项。
9. 如果选择了 **Customize database settings**，在 **JDBC Drive Configuration** 页面上指定数据源 JDBC 驱动程序厂商，请选择一个或多个驱动程序 JAR 文件，然后单击 **Next**。
数据源是一个对象，它允许 Java 数据库连接(JDBC)客户端（如应用服务器）与数据库建立连接。应用程序在 Java 命名和目录接口(JNDI)树或本地应用程序上下文中查找数据源，并请求数据库连接来检索数据。您必须为 KIE 服务器配置数据源，以确保在服务器和指定数据库之间进行正确的数据交换。
10. 如果您选择了 **Customize KIE 服务器属性属性**，在 **KIE Server Properties Configuration** 页面中更改以下任何属性：
 - 更改 **KIE 服务器 ID** 的值，以更改 KIE 服务器属性的名称。
 - 更改 **Controller URL** 的值，以更改无头进程 Automation Manager 控制器的 URL。
 - 取消选择您想禁用的任何 KIE 服务器功能。
11. 单击 **Next** 来配置运行时环境。
12. 当 **处理** 在屏幕顶部出现时，单击 **Next** 完成安装。
13. 可选：单击 **Generate Installation Script and Properties File** 将安装数据保存在 XML 文件中，然后点 **Done**。
安装程序会生成两个文件。**auto.xml** 文件可自动安装，**auto.xml.variables** 文件存储用户密码和其他敏感变量。使用 **auto.xml** 文件在具有相同类型服务器的多个系统中重复 Red Hat Process Automation Manager 安装，以及与原始安装相同的配置。如果需要，更新 **auto.xml** 文件中的 **installpath** 参数。要使用 XML 文件执行安装，请输入以下命令：

```
java -jar rhpam-installer-7.13.5.jar <path-to-auto.xml-file>
```

您已使用安装程序成功安装了 Red Hat Process Automation Manager。如果您只安装了 Business Central，请重复这些步骤以便在单独的服务器中安装 KIE 服务器。



注意

如果您使用 Microsoft SQL Server，请确保为您的数据库配置了适用的事务隔离。如果没有，您可能会遇到死锁。推荐的配置通过输入以下语句开启

ALLOW_SNAPSHOT_ISOLATION 和 **READ_COMMITTED_SNAPSHOT**：

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

13.2. 在 CLI 模式中使用安装程序

您可以使用命令行界面(CLI)来运行 Red Hat Process Automation Manager 安装程序。



注意

为安全起见，您应该以非 root 用户身份运行安装程序。

先决条件

- **Red Hat Process Automation Manager 7.13.5 安装程序**已下载。具体说明请查看 [第 12 章 下载 Red Hat Process Automation Manager 安装文件](#)。
- 已安装受支持的 JDK。有关支持的 JDK 列表，请参阅 [Red Hat Process Automation Manager 7 支持的配置](#)。
- 已备份的红帽 JBoss EAP 7.4 服务器安装可用。
- 需要足够的用户权限以完成安装。
- JAR 二进制文件包含在 **\$PATH** 环境变量中。在 Red Hat Enterprise Linux 上，它包含在 **java-\$JAVA_VERSION-openjdk-devel** 软件包中。



注意

Red Hat Process Automation Manager 旨在使用 UTF-8 编码。如果底层 JVM 使用了不同的编码系统，则可能发生意外错误。为确保 JVM 使用 UTF-8，请使用 **"-Dfile.encoding=UTF-8"** 系统属性。有关系统属性列表，请查看 [附录 A, Business Central 系统属性](#)。

流程

1. 在终端窗口中，进入下载安装程序文件的目录，并输入以下命令：

```
java -jar rhpam-installer-7.13.5.jar -console
```

命令行交互式过程将启动并显示最终用户许可证协议。

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

2. 阅读许可证协议，输入 **1**，然后按 **Enter** 继续：

```
Specify the home directory of one of the following servers: Red Hat JBoss EAP 7 or Red Hat JBoss Web Server 5. For more information, see
```

<https://access.redhat.com/articles/3405381>[Red Hat Process Automation Manager 7 Supported Configurations].

3. 输入现有 Red Hat JBoss EAP 7.4 安装的父目录。
安装程序将验证安装在提供的位置的位置。输入 **1** 确认并继续。



注意

您可以在同一服务器上安装 Business Central 和 KIE 服务器。然而，您应该在生产环境中在不同的服务器上安装 Business Central 和 KIE 服务器。

4. 按照安装程序中的说明完成安装。



注意

在创建用户名和密码时，请确保指定的用户名和密码与角色或组的任何已知标题不冲突。例如，如果有一个名为 **admin** 的角色，您不应创建用户名为 **admin** 的用户。

密码必须至少包含八个字符，且必须至少包含一个数字和一个非字母数字字符（不包括字符和）。

记录用户名和密码。您将需要他们访问 **Business Central** 和 **KIE 服务器**。

5. 安装完成后，您会看到此消息：

Would you like to generate an automatic installation script and properties file?

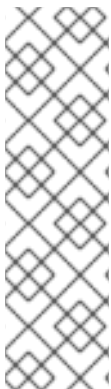
6. 输入 **y** 来创建包含安装数据的 XML 文件，或 **n** 完成安装。如果您输入 **y**，会提示您输入 XML 文件的路径。

7. 输入路径或者按 **Enter** 键接受推荐的路径。

安装程序会生成两个文件。**auto.xml** 文件可自动安装，**auto.xml.variables** 文件存储用户密码和其他敏感变量。在与原始安装相同的配置相同的服务器上，使用多个系统上的 **auto.xml** 文件轻松重复 Red Hat Process Automation Manager 安装。如果需要，更新 **auto.xml** 文件中的 **installpath** 参数。要使用 XML 文件执行安装，请输入以下命令：

```
java -jar rhpam-installer-7.13.5.jar <path-to-auto.xml-file>
```

8. 如果您只安装了 **Business Central**，请重复这些步骤以便在单独的服务器中安装 **KIE 服务器**。



注意

如果您使用 **Microsoft SQL Server**，请确保为您的数据库配置了适用的事务隔离。如果没有，您可能会遇到死锁。推荐的配置通过输入以下语句开启 **ALLOW_SNAPSHOT_ISOLATION** 和 **READ_COMMITTED_SNAPSHOT**：

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

13.3. 使用安装程序通过流程自动化管理器控制器配置 KIE 服务器

KIE 服务器可由流程自动化管理器控制器进行管理，也可以管理它。如果 **KIE 服务器**为非受管，您必须手动创建和维护 **KIE 容器**（部署单元）。如果管理 **KIE 服务器**，则“流程自动化管理器”控制器管理 **KIE 服务器配置**，并与流程自动化管理器控制器交互以创建和维护 **KIE 容器**。

流程自动化管理器控制器与 **Business Central** 集成。如果您安装 **Business Central**，您可以使用 **Business Central** 中的 **执行服务器** 页面与流程自动化管理器控制器交互。

您可以使用互动或 **CLI 模式**的安装程序来安装 **Business Central** 和 **KIE 服务器**，然后使用流程自动化管理器控制器配置 **KIE 服务器**。

先决条件

- 有两台已备份的 **Red Hat JBoss EAP 7.4** 服务器安装的计算机可用。
- 需要足够的用户权限以完成安装。

流程

1. 在第一个计算机上，以交互模式或 **CLI 模式**运行安装程序。如需更多信息，请参阅以下部分之一：
 - [第 13.1 节 “在互动模式中使用安装程序”](#)

- **第 13.2 节 “在 CLI 模式中使用安装程序”**

2. 在 **Component Selection** 页面中，清除 **KIE 服务器 框**。
3. 完成 **Business Central** 安装。
4. 在第二个计算机上，以交互模式或 **CLI 模式** 运行安装程序。
5. 在 **组件选择** 页面中，清除 **Business Central 框**。
6. 在 **Configure Runtime Environment** 页面中，选择 **Perform Advanced Configuration**。
7. 选择“自定义 **KIE 服务器**”属性，然后单击“下一步”。
8. 输入 **Business Central** 的控制器 URL，并为 **KIE 服务器** 配置其他属性。控制器 URL 具有以下格式，其中 `<HOST:PORT>` 是第二个计算机上的 **Business Central** 地址：

```
<HOST:PORT>/business-central/rest/controller
```
9. 完成安装。
10. 要验证流程自动化管理器控制器现在是否与 **Business Central** 集成，请转至 **Business Central** 中的 **执行服务器** 页面，并确认您配置的 **KIE 服务器** 是否出现在 **REMOTE SERVERS** 下。

第 14 章 通过 ZIP 文件安装 RED HAT PROCESS AUTOMATION MANAGER

您可以使用 Red Hat Process Automation Manager ZIP 文件（一个用于 Business Central，另一个用于 KIE 服务器）来安装 Red Hat Process Automation Manager，而无需使用安装程序置备的基础架构。



注意

您应该在生产环境中的不同服务器上安装 Business Central 和 KIE 服务器。

有关安装无头进程 Automation Manager 控制器的详情，请参考 [第 19 章 安装并运行无头进程自动化管理器控制器](#)。

14.1. 从 ZIP 文件安装 BUSINESS CENTRAL

Business Central 是用来创建和管理 KIE 服务器执行的业务规则的图形用户界面。您可以使用可部署的 ZIP 文件来安装和配置 Business Central。

先决条件

- 已备份的红帽 JBoss EAP 安装版本 7.4。红帽 JBoss EAP 安装的基础目录称为 *EAP_HOME*。
- 需要足够的用户权限以完成安装。
- 以下文件被下载，如 [第 12 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述：

`rhpm-7.13.5-business-central-eap7-deployable.zip`

流程

1. 将 `rhpm-7.13.5-business-central-eap7-deployable.zip` 文件提取到临时目录中。在以下示例中，此目录名为 *TEMP_DIR*。
2. 将 `TEMP_DIR/rhpm-7.13.5-central-eap7-deployable/jboss-eap-7.4` 目录的内容复制到

EAP_HOME。提示时，合并或替换文件。



警告

确定您复制的 Red Hat Process Automation Manager 部署的名称与您在 Red Hat JBoss EAP 实例中的现有部署没有冲突。

14.2. 从 ZIP 文件安装 KIE 服务器

KIE 服务器为业务资产提供运行时环境，并访问存储在资产存储库中的数据（知识库）。您可以使用可部署的 ZIP 文件来安装和配置 KIE 服务器。

先决条件

- 已备份的红帽 JBoss EAP 安装版本 7.4。红帽 JBoss EAP 安装的基础目录称为 **EAP_HOME**。
- 需要足够的用户权限以完成安装。
- 以下文件被下载，如 [第 12 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述：

rhpm-7.13.5-kie-server-ee8.zip

流程

1. 将 **rhpm-7.13.5-kie-server-ee8.zip** 存档提取到临时目录中。在以下示例中，此目录名为 **TEMP_DIR**。
2. 将 **TEMP_DIR/rhpm-7.13.5-kie-server-ee8/kie-server.war** 目录复制到 **EAP_HOME/standalone/deployments/**。

**警告**

确定您复制的 Red Hat Process Automation Manager 部署的名称不会与您在 Red Hat JBoss EAP 实例中的现有部署冲突。

3. 将 `TEMP_DIR/rhpam-7.13.5-kie-server-ee8/rhpam-7.13.5-kie-server-ee8/SecurityPolicy/` 的内容复制到 `EAP_HOME/bin`。
4. 当系统提示覆盖文件时，点 替换。
5. 在 `EAP_HOME/standalone/deployments/` 目录中，创建名为 `kie-server.war.dodeploy` 的空文件。此文件确保服务器启动时会自动部署 KIE 服务器。

**注意**

如果您使用 Microsoft SQL Server，请确保为您的数据库配置了适用的事务隔离。如果没有，您可能会遇到死锁。推荐的配置通过输入以下语句开启 `ALLOW_SNAPSHOT_ISOLATION` 和 `READ_COMMITTED_SNAPSHOT`：

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

14.3. 为 KIE 服务器配置 JDBC 数据源

数据源是一个对象，它允许 Java 数据库连接(JDBC)客户端（如应用服务器）与数据库建立连接。应用程序在 Java 命名和目录接口(JNDI)树或本地应用程序上下文中查找数据源，并请求数据库连接来检索数据。您必须为 KIE 服务器配置数据源，以确保在服务器和指定数据库之间进行正确的数据交换。

通常，使用 Red Hat Process Automation Manager 的解决方案在单个事务中管理多个资源。用于异步作业、事件和计时器的 JMS，例如：在确保数据原子性和一致性结果时，Red Hat Process Automation Manager 需要数据源中的 XA 驱动程序。如果监听器内存在针对不同架构的事务代码，或者从 jBPM 引擎提供的 hook 派生出来，则还需要一个 XA 驱动程序。

不要使用非 XA 数据源，除非您正面没有参与单一事务的多个资源。



注意

对于生产环境，请指定实际的数据源。不要在生产环境中使用示例数据源。

先决条件

- 您希望用于创建数据库连接的 JDBC 供应商在您要在其上部署 KIE 服务器的所有服务器上配置，如红帽 *JBoss 企业应用服务器配置指南* 的 "Creating Datasources" 和 "JDBC 驱动程序" 部分中所述。
- Red Hat Process Automation Manager 7.13.5 Add Ons (rhpam-7.13.5-add-ons.zip) 文件从红帽客户门户中的 [软件下载页面](#) 下载。

流程

1. 完成以下步骤以准备数据库：
 - a. 在临时目录中提取 rhpam-7.13.5-add-ons.zip，如 TEMP_DIR。
 - b. 提取 TEMP_DIR/rhpam-7.13.5-migration-tool.zip。
 - c. 将您的当前目录更改为 TEMP_DIR/rhpam-7.13.5-migration-tool/ddl-scripts 目录。此目录包含多种数据库类型的 DDL 脚本。
 - d. 将您的数据库类型的 DDL 脚本导入到您要使用的数据库。

以下示例在 PostgreSQL 中创建 jBPM 数据库结构

```
psql jbpm < /ddl-scripts/postgresql/postgresql-jbpm-schema.sql
```



注意

如果您与 Spring Boot 结合使用 PostgreSQL 或 Oracle，则必须导入对应的 Spring Boot DDL 脚本，如 /ddl-scripts/oracle/oracle-springboot-jbpm-schema.sql 或 /ddl-scripts/postgresql/postgresql-springboot-jbpm-schema.sql。



注意

PostgreSQL DDL 脚本创建 PostgreSQL 模式，其自动递增带有 @LOB 的实体属性的整数值(OID)列。要使用其他二进制列，如 BYTEA 而不是 OID，您必须使用 `postgresql-bytea-jbpm-schema.sql` 脚本创建 PostgreSQL 模式，并设置 Red Hat Process Automation Manager `org.kie.persistence.postgresql.useText=true` 和 `org.kie.persistence.postgresql.useBytea=true` 标志。在创建基于 BYTEA 的 schema 时，不要使用 `postgresql-jbpm-lo-trigger-clob.sql` 脚本。Red Hat Process Automation Manager 不提供从基于 OID 到基于 BYTEA 的 schema 的迁移工具。

2. 在文本编辑器中打开 `EAP_HOME/standalone/configuration/standalone-full.xml`，并找到 `<system-properties>` 标签。
3. 在 `<system-properties>` 标签中添加以下属性，其中 `<DATASOURCE>` 是数据源的 JNDI 名称，`<HIBERNATE_DIALECT>` 是您的数据库休眠状态。



注意

`org.kie.server.persistence.ds` 属性的默认值是 `java:jboss/datasources/ExampleDS`。`org.kie.server.persistence.dialect` 属性的默认值是 `org.hibernate.dialect.H2Dialect`。

```
<property name="org.kie.server.persistence.ds" value="<DATASOURCE>"/>
<property name="org.kie.server.persistence.dialect" value="<HIBERNATE_DIALECT>"/>
```

以下示例演示了如何为 PostgreSQL hibernate dialect 配置数据源：

```
<system-properties>
  <property name="org.kie.server.repo" value="${jboss.server.data.dir}"/>
  <property name="org.kie.example" value="true"/>
  <property name="org.jbpm.designer.perspective" value="full"/>
  <property name="designerdataobjects" value="false"/>
  <property name="org.kie.server.user" value="rhpamUser"/>
  <property name="org.kie.server.pwd" value="rhpam123!"/>
  <property name="org.kie.server.location" value="http://localhost:8080/kie-
server/services/rest/server"/>
  <property name="org.kie.server.controller" value="http://localhost:8080/business-
central/rest/controller"/>
  <property name="org.kie.server.controller.user" value="kieserver"/>
  <property name="org.kie.server.controller.pwd" value="kieserver1!"/>
  <property name="org.kie.server.id" value="local-server-123"/>
```

```
<!-- Data source properties. -->
<property name="org.kie.server.persistence.ds"
value="java:jboss/datasources/KieServerDS"/>
<property name="org.kie.server.persistence.dialect"
value="org.hibernate.dialect.PostgreSQLDialect"/>
</system-properties>
```

支持以下分区：

- **DB2:** org.hibernate.dialect.DB2Dialect
- **MSSQL:** org.hibernate.SQLServer 2012Dialect
- **MySQL:** org.hibernate.dialect.MySQL5InnoDBDialect
- **MariaDB :** org.hibernate.dialect.MySQL5InnoDBDialect
- **Oracle :** org.hibernate.dialect.Oracle10gDialect
- **PostgreSQL :** org.hibernate.dialect.PostgreSQL82Dialect
- **PostgreSQL 加上 :** org.hibernate.dialect.PostgresPlusPlusDialect
- **Sybase:** org.hibernate.dialect.SybaseASE157Dialect

14.4. 创建用户

如果您使用可部署的 ZIP 文件来安装 Red Hat Process Automation Manager，则才能登录到 Business Central 或 KIE 服务器，您必须创建用户。本节演示了如何使用 admin、rest-all 和 kie-server 角色创建 Business Central 用户，以及 kie-server 角色的 KIE 服务器用户。有关角色的详情，请参考 [第 11 章 Red Hat Process Automation Manager 角色和用户](#)。



注意

Red Hat Process Automation Manager 将用户数据作为一组属性或一组文件来存储。基于文件的存储提供了几个额外的功能，如 **SSH 登录**和**用户维护 UI**。

本文档中的用户脚本示例使用基于文件的用户脚本 `jboss-cli.sh`，而不是基于属性的用户脚本 `add-users.sh`。

先决条件

- **Red Hat Process Automation Manager** 安装在 **Red Hat JBoss EAP** 安装(`EAP_HOME`)的 **Base** 目录中。

流程

1. 可选：要使用基于属性的用户存储来更改 **Red Hat Process Automation Manager**，请完成以下步骤：
 - a. 运行以下命令以应用 `kie-fs-realm` 补丁：

```
$. /bin/elytron-tool.sh filesystem-realm --users-file application-users.properties --roles-file application-roles.properties --output-location kie-fs-realm-users
```
 - b. 打开每个 `kie-fs-realm-users/*/<USER>.xml` 文件，其中 `<USER>` 是一个 **Red Hat Process Automation Manager** 用户。
 - c. 将 `<attribute name="roles" value=` 替换为 `<attribute name="role" value=`。
2. 在终端应用中，导航到 `EAP_HOME/bin` 目录。
3. 使用 `admin`、`rest-all` 和 `kie-server` 角色创建一个用户，以登录 **Business Central**。



注意

具有管理员角色的用户是 **Business Central** 管理员。具有 **rest-all** 角色的用户可以访问 **Business Central REST** 功能。具有 **kie-server** 角色的用户可以访问 **KIE 服务器 (KIE 服务器) REST** 功能。**kie-server** 角色是用户必须有权访问 **Business Central** 中的 **管理和追踪视图**。

在以下命令中，将 `< USERNAME >` 和 `< PASSWORD >` 替换为您选择的用户名和密码：

```
$ ./bin/jboss-cli.sh --commands="embed-server --std-out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>),/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password=<PASSWORD>}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=[admin,rest-all,kie-server])"
```



注意

确保指定的用户名与现有用户、角色或组不同。例如，不要创建用户名为 **admin** 的用户。

密码必须至少包含八个字符，且必须至少包含一个数字和一个非字母数字字符，但不包括 **和 (ampersand)**。

4.

使用 **kie-server** 角色创建一个用户，用于登录到 **KIE 服务器**。

```
$ ./bin/jboss-cli.sh --commands="embed-server --std-out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>),/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=[kie-server])"
```

5.

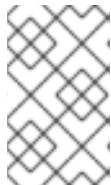
记录您的用户名和密码。

6.

可选：如果您在同一服务器实例中安装 **Business Central** 和 **KIE 服务器**，您可以创建一个具有这两个角色的单一用户：

```
$ ./bin/jboss-cli.sh --commands="embed-server --std-
```

```
out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>)/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>'}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=[admin,rest-all,kie-server])"
```



注意

您应该在生产环境中的不同服务器上安装 **Business Central** 和 **KIE 服务器**。

7.

可选：要一次创建多个用户，请创建一个包含用户数据并运行以下命令的文件，其中 `<USER_DATA>.cli` 是包含用户数据的文件：

```
$ ./bin/jboss-cli.sh --file=<USER_DATA>.cli
```

& It;USER_DATA>.cli 文件应包含类似以下示例的数据：

```
embed-server --std-out=echo

# first user
/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=
<USERNAME>)
/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=
<USERNAME>, clear={password='<PASSWORD>'})
/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-
attribute(identity=<USERNAME>, name=role, value=[admin,role,group])

# second user
...
```

14.5. 配置 KIE 服务器以连接至 BUSINESS CENTRAL



警告

本节提供了一个示例设置，可用于测试目的。部分值不适用于生产环境，并被标记为。

如果您的红帽流程自动化管理器环境中没有配置 **KIE 服务器**，或者需要红帽流程自动化管理器环境中

的其他 KIE 服务器，您必须配置 KIE 服务器以连接到 Business Central。



注意

如果要在 Red Hat OpenShift Container Platform 上部署 KIE 服务器，请参阅使用 [Operator 在 Red Hat OpenShift Container Platform 4 上部署 Red Hat Process Automation Manager 环境](#)。有关将其配置为连接到 Business Central 的说明。

先决条件

- Business Central 和 KIE 服务器安装在 Red Hat JBoss EAP 安装(*EAP_HOME*)的基础目录中，如以下部分所述：
 - [第 14.1 节 “从 ZIP 文件安装 Business Central”](#)
 - [第 14.2 节 “从 ZIP 文件安装 KIE 服务器”](#)
- 存在具有以下角色的用户：
 - 在 Business Central 中，拥有 其余角色的用户
 - 在 KIE 服务器上，角色为 `kie-server`的用户

更多信息请参阅 [第 14.4 节 “创建用户”](#)。

流程

1. 在 Red Hat Process Automation Manager 安装目录中，导航至 `standalone-full.xml` 文件。例如，如果您使用红帽 JBoss EAP 安装红帽流程自动化管理器，请转至 `$EAP_HOME/standalone/configuration/standalone-full.xml`。
2. 打开 `standalone-full.xml` 文件并在 `< system-properties>` 标签下设置以下 JVM 属性：

表 14.1. KIE 服务器实例的 JVM 属性

属性	值	备注
org.kie.server.id	default-kie-server	KIE 服务器 ID。
org.kie.server.controller	http://localhost:8080/business-central/rest/controller	Business Central 的位置。连接到 Business Central API 的 URL。
org.kie.server.controller.user	controllerUser	用户名（具有 rest-all 角色）可以登录到 Business Central。
org.kie.server.controller.pwd	controllerUser1234;	登录 Business Central 的用户的密码。
org.kie.server.location	http://localhost:8080/kie-server/services/rest/server	KIE 服务器的位置。连接到 KIE 服务器的 API 的 URL。

表 14.2. Business Central 实例的 JVM 属性

属性	值	备注
org.kie.server.user	controllerUser	使用角色 kie-server 的用户名。
org.kie.server.pwd	controllerUser1234;	用户的密码。

以下示例演示了如何配置 KIE 服务器实例：

```
<property name="org.kie.server.id" value="default-kie-server"/>
<property name="org.kie.server.controller" value="http://localhost:8080/business-central/rest/controller"/>
<property name="org.kie.server.controller.user" value="controllerUser"/>
<property name="org.kie.server.controller.pwd" value="controllerUser1234;"/>
<property name="org.kie.server.location" value="http://localhost:8080/kie-server/services/rest/server"/>
```

以下示例演示了如何为 Business Central 实例配置：

```
<property name="org.kie.server.user" value="controllerUser"/>
<property name="org.kie.server.pwd" value="controllerUser1234;"/>
```

3.

要验证 KIE 服务器是否成功启动，请在 KIE 服务器运行时向 `http://SERVER:PORT/kie-server/services/rest/server/` 发送 GET 请求。有关在 KIE 服务器上运行 Red Hat Process Automation Manager 的更多信息，请参阅 [运行 Red Hat Process Automation Manager](#)。

身份验证成功后，您会收到类似以下示例的 XML 响应：

```
<response type="SUCCESS" msg="Kie Server info">
  <kie-server-info>
    <capabilities>KieServer</capabilities>
    <capabilities>BRM</capabilities>
    <capabilities>BPM</capabilities>
    <capabilities>CaseMgmt</capabilities>
    <capabilities>BPM-UI</capabilities>
    <capabilities>BRP</capabilities>
    <capabilities>DMN</capabilities>
    <capabilities>Swagger</capabilities>
    <location>http://localhost:8230/kie-server/services/rest/server</location>
    <messages>
      <content>Server KieServerInfo{serverId='first-kie-server', version='7.5.1.Final-redhat-1', location='http://localhost:8230/kie-server/services/rest/server', capabilities=[KieServer, BRM, BPM, CaseMgmt, BPM-UI, BRP, DMN, Swagger]}started successfully at Mon Feb 05 15:44:35 AEST 2018</content>
      <severity>INFO</severity>
      <timestamp>2018-02-05T15:44:35.355+10:00</timestamp>
    </messages>
    <name>first-kie-server</name>
    <id>first-kie-server</id>
    <version>7.5.1.Final-redhat-1</version>
  </kie-server-info>
</response>
```

4.

验证成功注册：

a.

登录到 **Business Central**。

b.

点 **Menu** → **Deploy** → **Execution Servers**。

如果注册成功，您将看到注册的服务器 ID。

14.6. 线程效率

要确保使用最佳线程数，请将线程系统属性的值设置为 CPU 数加上一。

1.

在 Red Hat Process Automation Manager 安装目录中，导航至 `standalone-full.xml` 文件。例如，如果您使用红帽 JBoss EAP 安装红帽流程自动化管理器，请转至 `$EAP_HOME/standalone/configuration/standalone-full.xml`。

2. 打开 `standalone-full.xml` 文件。
3. 在 `<system-properties>` 标签下，将以下 JVM 属性的值设置为 CPU 数量加上一个：

```
org.appformer.concurrent.managed.thread.limit  
org.appformer.concurrent.unmanaged.thread.limit  
org.appformer.concurrent.indexing.thread.limit  
org.appformer.concurrent.rest.api.thread.limit
```



注意

CPU 的数量加上一个是所有属性的有效基准值。您可能必须根据其他测试进行微调。

第 15 章 使用密钥存储保护密码

您可以使用密钥存储来加密用于 Business Central 和 KIE 服务器之间的通信的密码。您应该同时对控制器和 KIE 服务器密码进行加密。如果 Business Central 和 KIE 服务器部署到不同的应用程序服务器，则两个应用服务器都应该使用密钥存储。

为您的密钥存储使用 Java Cryptography Extension KeyStore(JCEKS)，因为它支持对称密钥。使用 KeyTool 作为 JDK 安装的一部分，可以创建一个新的 JCEKS。



注意

如果没有使用 JCEKS 配置 KIE 服务器，则 KIE 服务器密码以纯文本形式保存在系统属性中。

先决条件

- KIE 服务器安装在红帽 JBoss EAP 中。
- 已安装 Java 8 或更高版本。

流程

1.

在 Red Hat JBoss EAP 主目录中，输入以下命令创建带有 kie-server 角色的 KIE 服务器用户并指定密码。在以下示例中，将 `<USERNAME>` 和 `<PASSWORD>` 替换为您选择的用户名和密码。

```
$<EAP_HOME>./bin/jboss-cli.sh --commands="embed-server --std-out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>),/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>'}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=['kie-server'])"
```

2.

要使用 KeyTool 创建 JCEKS，请在 Java 8 主目录中输入以下命令：

```
$<JAVA_HOME>/bin/keytool -importpassword -keystore <KEYSTORE_PATH> -keypass <ALIAS_KEY_PASSWORD> -alias <PASSWORD_ALIAS> -storepass <KEYSTORE_PASSWORD> -storetype JCEKS
```

在本例中，替换以下变量：

- `<KEYSTORE_PATH >` : 存储密钥存储的路径
 - `<KEYSTORE_PASSWORD>` : 密钥存储密码
 - `<ALIAS_KEY_PASSWORD >` : 用于访问使用别名存储的值的密码
 - `<PASSWORD_ALIAS >` : 条目的别名
3. 在提示时，输入您创建的 KIE 服务器用户的密码。
4. 在 `EAP_HOME/standalone/configuration/standalone-full.xml` 文件中设置以下系统属性，并替换下表中所列的占位符：

```
<system-properties>
  <property name="kie.keystore.keyStoreURL" value="<KEYSTORE_URL>"/>
  <property name="kie.keystore.keyStorePwd" value="<KEYSTORE_PWD>"/>
  <property name="kie.keystore.key.server.alias" value="<KEY_SERVER_ALIAS>"/>
  <property name="kie.keystore.key.server.pwd" value="<KEY_SERVER_PWD>"/>
  <property name="kie.keystore.key.ctrl.alias" value="<KEY_CONTROL_ALIAS>"/>
  <property name="kie.keystore.key.ctrl.pwd" value="<KEY_CONTROL_PWD>"/>
</system-properties>
```

表 15.1. 用于加载 KIE 服务器 JCEKS 的系统属性

系统属性	占位符	描述
<code>kie.keystore.keyStoreURL</code>	<code><KEYSTORE_URL></code>	要使用的 JCEKS 的 URL，如 file:///home/kie/keystores/keystore.jceks
<code>kie.keystore.keyStorePwd</code>	<code><KEYSTORE_PWD></code>	JCEKS 的密码
<code>kie.keystore.key.server.alias</code>	<code><KEY_SERVER_ALIASS></code>	保存密码的 REST 服务的密钥别名
<code>kie.keystore.key.server.pwd</code>	<code><KEY_SERVER_PWD></code>	使用存储的密码的 REST 服务别名的密码

系统属性	占位符	描述
kie.keystore.key.ctrl.alias	<KEY_CONTROL_AL IAS>	存储密码的默认 REST Process Automation Controller 的密钥别名
kie.keystore.key.ctrl.pwd	<KEY_CONTROL_P WD>	默认 REST Process Automation Controller 的别名密码，其存储的密码

5.

启动 KIE 服务器以验证配置。

第 16 章 将 SSH 配置为使用 RSA

SSH 用于克隆 Git 存储库。默认情况下，Business Central 提供 DSA 加密算法。但是，一些 SSH 客户端（如 Fedora 23 环境中的 SSH 客户端）使用 RSA 算法而不是 DSA 算法。Business Central 包含一个系统属性，可用于在需要时从 DSA 切换到 RSA。



注意

支持的配置（如 Red Hat Enterprise Linux 7）上的 SSH 客户端不受此问题的影响。有关支持的配置列表，请参阅 [Red Hat Process Automation Manager 7 支持的配置](#)。

流程

完成以下任务之一以启用这个系统属性：

- 按如下方式修改客户端上的 `~/.ssh/config` 文件，以强制 SSH 客户端接受已弃用的 DSA 算法：

```
Host <SERVER_IP>
  HostKeyAlgorithms +ssh-dss
```

- 开始 Business Central 时包括 `-Dorg.uberfire.nio.git.ssh.algorithm=RSA` 参数，例如：

```
$ ./standalone.sh -c standalone-full.xml
-Dorg.uberfire.nio.git.ssh.algorithm=RSA
```

第 17 章 验证 RED HAT PROCESS AUTOMATION MANAGER 安装

安装 Red Hat Process Automation Manager 后，创建一个资产来验证安装是否正常工作。

流程

1. 在终端窗口中，导航到 `EAP_HOME/bin` 目录，再输入以下命令来启动 Red Hat Process Automation Manager：

```
./standalone.sh -c standalone-full.xml
```



注意

如果您在没有 KIE 服务器的情况下部署了 Business Central，则可使用 `standalone.sh` 脚本启动 Business Central，而不指定 `standalone-full.xml` 文件。在这种情况下，请确保在启动 Business Central 前对 `standalone.xml` 文件进行任何更改。

2. 在 Web 浏览器中，输入 `localhost:8080/business-central`。
 - 如果您将 Red Hat Process Automation Manager 配置为从域名运行，请使用域名替换 `localhost`，例如：

```
http://www.example.com:8080/business-central
```

- 如果您将 Red Hat Process Automation Manager 配置为在集群中运行，请使用特定节点的 IP 地址替换 `localhost`，例如：

```
http://<node_IP_address>:8080/business-central
```

3. 输入您在安装过程中创建的 admin 用户凭证。此时会出现 Business Central 主页。
4. 选择 **Menu** → **Deploy** → **Execution Servers**。

5. 确认 **default-kieserver** 已列在 **Server Configuration** 下。
6. 选择 **Menu** → **Design** → **Projects**。
7. 打开 **MySpace** 空间。
8. 点 **Try Samples** → **Mortgages** → **OK**。此时会出现 **Assets** 窗口。
9. 点 **Add Asset** → **Data Object**。
10. 在 **Data Object** 字段中输入 **MyDataObject**，然后单击 **OK**。
11. 点 **Spaces** → **MySpace** → **Mortgages** 并确认 **MyDataObject** 位于资产列表中。
12. 删除 **Mortgages** 项目。

第 18 章 运行 RED HAT PROCESS AUTOMATION MANAGER

使用此流程，以单机模式在 Red Hat JBoss EAP 上运行 Red Hat Process Automation Manager。

先决条件

- Red Hat Process Automation Manager 已安装并配置。



注意

如果您更改了默认主机(localhost)或默认端口(9990)，那么在运行 Red Hat Process Automation Manager 前，您必须编辑 Business -central.war/WEB-INF/classes/datasource-management.properties 和 Business -central.war/WEB-INF/security-management.properties 文件，如 [35 Solution1 955](#)。

流程

1. 在终端应用中，导航到 *EAP_HOME/bin*。

2. 运行独立配置：

- 在 Linux 或基于 UNIX 的系统中：

```
$. /standalone.sh -c standalone-full.xml
```

- 在 Windows 中：

```
standalone.bat -c standalone-full.xml
```



注意

如果您在没有 KIE 服务器的情况下部署了 **Business Central**，则可使用 **standalone.sh** 脚本启动 **Business Central**，而不指定 **standalone-full.xml** 文件。在这种情况下，请确保在启动 **Business Central** 前对 **standalone.xml** 文件进行任何更改。

在 Linux 或基于 UNIX 的系统中：

```
┆ $ ./standalone.sh
```

在 Windows 中：

```
┆ standalone.bat
```

3. 在网页浏览器中，打开 URL **localhost:8080/business-central**。

如果您将 **Red Hat Process Automation Manager** 配置为从域名运行，请使用域名替换 **localhost**，例如：

http://www.example.com:8080/business-central

4. 使用您在 [第 14.4 节“创建用户”](#) 中为 **Business Central** 创建的用户凭证登录。

第 19 章 安装并运行无头进程自动化管理器控制器

您可以将 KIE 服务器配置为在受管或非受管模式下运行。如果 KIE 服务器为非受管，您必须手动创建和维护 KIE 容器（部署单元）。如果管理 KIE 服务器，则“流程自动化管理器”控制器管理 KIE 服务器配置，并与流程自动化管理器控制器交互以创建和维护 KIE 容器。

Business Central 具有一个嵌入式流程自动化管理器控制器。如果您安装 **Business Central**，请使用 **执行服务器** 页面来创建和维护 KIE 容器。如果要在没有 **Business Central** 的情况下自动执行 KIE 服务器管理，您可以使用无头流程自动化管理器控制器。

19.1. 安装无头进程自动化管理器控制器

您可以安装无头进程自动化管理器控制器，并使用 REST API 或 KIE 服务器 Java 客户端 API 与它交互。

先决条件

- 已备份的红帽 JBoss EAP 安装版本 7.4。红帽 JBoss EAP 安装的基础目录称为 **EAP_HOME**。
- 需要足够的用户权限以完成安装。

流程

1. 进入红帽客户门户网站中的 **Software Downloads** 页面（需要登录），然后从下拉列表中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.13.5
2. 下载 Red Hat Process Automation Manager 7.13.5 Add Ons (rhpam-7.13.5-add-ons.zip 文件)。
3. 提取 rhpam-7.13.5-add-ons.zip 文件。rhpam-7.13.5-controller-ee7.zip 文件位于提取的目录中。

4. 将 `rhcam-7.13.5-controller-ee7.zip` 存档提取到临时目录中。在以下示例中，此目录名为 `TEMP_DIR`。
5. 将 `TEMP_DIR/rhcam-7.13.5-controller-ee7/controller.war` 目录复制到 `EAP_HOME/standalone/deployments/`。



警告

确保您复制的无头进程自动化管理器控制器部署的名称与您在 Red Hat JBoss EAP 实例中的现有部署不冲突。

6. 将 `TEMP_DIR/rhcam-7.13.5-controller-ee7/SecurityPolicy/` 目录的内容复制到 `EAP_HOME/bin`。
7. 当系统提示覆盖文件时，选择是。
8. 在 `EAP_HOME/standalone/deployments/` 目录中，创建名为 `controller.war.dodeploy` 的空文件。此文件确保服务器启动时自动部署无头进程自动化管理器控制器。

19.1.1. 创建无头进程自动化管理器控制器用户

在使用无头进程 Automation Manager 控制器前，您必须创建一个具有 `kie-server` 角色的用户。

先决条件

- 无头进程自动化管理器控制器安装在 Red Hat JBoss EAP 安装(`EAP_HOME`)的 Base 目录中。

流程

1. 在终端应用中，导航到 `EAP_HOME/bin` 目录。
2. 输入以下命令并将 `< USERNAME >` 和 `< PASSWORD >` 替换为您选择的用户名和密码。

```

$ ./bin/jboss-cli.sh --commands="embed-server --std-
out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-
identity(identity=<USERNAME>)/subsystem=elytron/filesystem-
realm=ApplicationRealm:set-password(identity=<USERNAME>, clear=
{password='<PASSWORD>')/subsystem=elytron/filesystem-
realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role,
value=['kie-server'])"

```



注意

确保指定的用户名与现有用户、角色或组不同。例如，不要创建用户名为 `admin` 的用户。

密码必须至少包含八个字符，且必须至少包含一个数字和一个非字母数字字符，但不包括 `和 (ampersand)`。

3. 记录您的用户名和密码。

19.1.2. 配置 KIE 服务器和无头进程自动化管理器控制器

如果 KIE 服务器由无头进程自动化管理器控制器管理，您必须编辑 KIE 服务器安装中的 `standalone-full.xml` 文件，并在无头进程自动化管理器安装中编辑 `standalone.xml` 文件。

先决条件

- KIE 服务器安装在 `EAP_HOME` 中。
- 无头进程自动化管理器控制器安装在 `EAP_HOME` 中。



注意

您应该在生产环境中的不同服务器上安装 KIE 服务器和无头进程自动化管理器控制器。但是，如果您在同一服务器上安装 KIE 服务器和无头进程自动化管理器控制器，例如在开发环境中，请在共享的 `standalone-full.xml` 文件中进行这些更改。

- 在 KIE 服务器节点上，存在具有 `kie-server` 角色的用户。
- 在服务器节点上，存在具有 `kie-server` 角色的用户。

更多信息请参阅 [第 14.4 节“创建用户”](#)。

流程

1. 在 `EAP_HOME/standalone/configuration/standalone-full.xml` 文件中，将以下属性添加到 `< system-properties>` 部分，并将 `< USERNAME >` 和 `< USER_PWD >` 替换为该用户的凭证：

```
<property name="org.kie.server.user" value="<USERNAME>"/>
<property name="org.kie.server.pwd" value="<USER_PWD>"/>
```

2. 在 KIE Server `EAP_HOME/standalone/configuration/standalone-full.xml` 文件中，将以下属性添加到 `< system-properties>` 部分：

```
<property name="org.kie.server.controller.user" value="<CONTROLLER_USER>"/>
<property name="org.kie.server.controller.pwd" value="<CONTROLLER_PWD>"/>
<property name="org.kie.server.id" value="<KIE_SERVER_ID>"/>
<property name="org.kie.server.location" value="http://<HOST>:<PORT>/kie-server/services/rest/server"/>
<property name="org.kie.server.controller" value="<CONTROLLER_URL>"/>
```

3. 在这个文件中，替换以下值：

- 将 `< CONTROLLER_USER >` 和 `< CONTROLLER_PWD >` 替换为用户的凭证，并将 `kie-server` 角色替换为 `kie-server` 角色。
- 将 `< KIE_SERVER_ID >` 替换为 KIE 服务器安装的 ID 或名称，例如 `rhcam-7.13.5-kie-server-1`。

- 将 `<HOST >` 替换为 KIE 服务器主机的 ID 或名称，例如 `localhost` 或 `192.7.8.9`。
- 将 `<PORT >` 替换为 KIE 服务器主机的端口，例如 `8080`。



注意

`org.kie.server.location` 属性指定 KIE 服务器的位置。

- 将 `<CONTROLLER_URL >` 替换为无头进程 Automation Manager 控制器的 URL。
KIE 服务器在启动过程中连接到此 URL。

19.2. 运行无头进程自动化管理器控制器

在 Red Hat JBoss EAP 上安装了无头进程自动化管理器控制器后，使用此流程运行无头进程自动化管理器控制器。

先决条件

- 无头进程自动化管理器控制器会在 Red Hat JBoss EAP 安装(`EAP_HOME`)的 Base 目录中安装和配置。

流程

1. 在终端应用中，导航到 `EAP_HOME/bin`。
2. 如果您在与安装 KIE 服务器的 Red Hat JBoss EAP 实例相同的 Red Hat JBoss EAP 实例中安装无头进程自动化管理器控制器，请输入以下命令：

- 在 Linux 或基于 UNIX 的系统中：

```
$ ./standalone.sh -c standalone-full.xml
```

- 在 Windows 中：

```
standalone.bat -c standalone-full.xml
```

3.

如果您在安装了 KIE 服务器的红帽 JBoss EAP 实例中安装了无头流程自动化管理器控制器，请使用 `standalone.sh` 脚本启动无头进程自动化管理器控制器：



注意

在这种情况下，请确保对 `standalone.xml` 文件进行所有必要的配置更改。

- 在 Linux 或基于 UNIX 的系统中：

```
$./standalone.sh
```

- 在 Windows 中：

```
standalone.bat
```

4.

要验证无头进程自动化管理器控制器是否在使用 Red Hat JBoss EAP，请输入以下命令 `<CONTROLLER>` 和 `<CONTROLLER_PWD>` 是您在 [第 19.1.1 节“创建无头进程自动化管理器控制器用户”](#) 中创建的用户名和密码组合。此命令的输出提供有关 KIE 服务器实例的信息。

```
curl -X GET "http://<HOST>:<PORT>/controller/rest/controller/management/servers" -H "accept: application/xml" -u '<CONTROLLER>:<CONTROLLER_PWD>'
```



注意

另外，您可以使用 KIE 服务器 Java API 客户端访问无头进程自动化管理器控制器。

第 20 章 配置并运行独立 BUSINESS CENTRAL

您可以使用 Business Central 独立 JAR 文件运行 Business Central，而无需将其部署到应用服务器。您可以使用示例配置文件来启动 Business Central 独立 JAR 文件，也可以根据您的要求自定义 `sampfiles`。

**注意**

只有在 Red Hat Enterprise Linux 上运行时，才支持这个 JAR 文件。

先决条件

- Red Hat Process Automation Manager 7.13.5 Business Central Standalone (`rhpm-7.13.5-business-central-standalone.jar`)和 Red Hat Process Automation Manager 7.13.5 Add Ons (`rhpm-7.13.5-add-ons.zip`)文件已从 Red Hat Process Automation Manager 7.13 的 [Software Downloads](#) 页面下载，如 [第 12 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述。

流程

1. 将下载的 `rhpm-7.13.5-addons.zip` 提取到临时目录中。此存档包括 `rhpm-7.13.5-standalone-sample-configuration.zip` 文件。
2. 将 `rhpm-7.13.5-standalone-sample-configuration.zip` 文件提取到包含 `rhpm-7.13.5-business-central-standalone.jar` 文件的目录。`rhpm-7.13.5-standalone-sample-configuration.zip` 文件包含以下示例配置文件：

- `application-script.cli`: 添加用户和 kie 服务器系统属性的 Sample 脚本
- `kie-fs-realm-users`: Sample user data

您可以使用配置文件中提供的示例数据运行 `rhpm-7.13.5-business-central-standalone.jar` 文件，或者您可以根据您的要求自定义数据。

3. 要自定义配置数据，请完成以下步骤：
 - a. 编辑 `application-script.cli` 文件，使其包含具有 `admin`、`用户`、`rest-all`、`rest-client`

和 `kie-server` 角色的管理用户。在以下示例中，将 `<USERNAME>` 和 `<PASSWORD>` 替换为您要创建的用户的用户名和密码。

```
/subsystem=elytron/filesystem-realm=KieRealm:add-identity(identity=<USERNAME>)
/subsystem=elytron/filesystem-realm=KieRealm:set-password(identity=<USERNAME>,
clear={password="<PASSWORD>"})
/subsystem=elytron/filesystem-realm=KieRealm:add-identity-attribute(identity=
<USERNAME>, name=role, value=["admin","user","rest-all","rest-client","kie-server"])
```

4.

要运行 **Business Central** 独立 JAR 文件，请输入以下命令：

```
java -jar rhpam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli
```

5.

要在运行 JAR 文件时设置应用程序属性，请在命令中包含 `-D<PROPERTY>=<VALUE>`；参数，其中 `<PROPERTY>` 是受支持的应用程序属性的名称，`<VALUE>` 是属性值：

```
java -jar rhpam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli -
D<PROPERTY>=<VALUE> -D<PROPERTY>=<VALUE>
```

例如，要运行 **Business Central** 并以用户 `controllerUser` 身份连接到 KIE 服务器，请输入：

```
java -jar rhpam-7.13.5-business-central-standalone.jar \
--cli-script=application-script.cli \
-Dorg.kie.server.user=controllerUser \
-Dorg.kie.server.pwd=controllerUser1234
```

这样做可让您将容器部署到 KIE 服务器。如需更多信息，请参阅 [附录 A, Business Central 系统属性](#)。



注意

要在 **Business Central** 中启用用户和组管理，将 `org.uberfire.ext.security.management.wildfly.cli.folderPath` 属性的值设置为 `kie-fs-realm-users`。

第 21 章 RED HAT PROCESS AUTOMATION MANAGER 的 MAVEN 设置和软件仓库

创建 Red Hat Process Automation Manager 项目时，Business Central 使用为 Business Central 配置的 Maven 存储库。您可以通过修改 Maven 项目对象模型(POM)文件(pom.xml)文件，使用 Maven 全局或用户设置来指示所有 Red Hat Process Automation Manager 项目，从公共 Red Hat Process Automation Manager 存储库检索依赖项。您还可以将 Business Central 和 KIE 服务器配置为使用外部 Maven 存储库，或准备 Maven mirror 以供离线使用。

有关 Red Hat Process Automation Manager 打包和部署选项的更多信息，请参阅 [打包和部署 Red Hat Process Automation Manager 项目](#)。

21.1. 为 RED HAT PROCESS AUTOMATION MANAGER 添加 MAVEN 依赖项

要在 Red Hat Process Automation Manager 项目中使用正确的 Maven 依赖项，请将 Red Hat Business Automation 材料(BOM)文件添加到项目的 pom.xml 文件中。Red Hat Business Automation BOM 适用于 Red Hat Decision Manager 和 Red Hat Process Automation Manager。添加 BOM 文件时，项目中包含来自提供的 Maven 存储库的正确依赖项版本。

有关 Red Hat Automation BOM 的更多信息，请参阅 [Red Hat Process Automation Manager 和 Maven 库版本之间的映射是什么？](#)

流程

1. 在 pom.xml 文件中声明 Red Hat Business Automation BOM :

```
<dependencyManagement>
<dependencies>
<dependency>
<groupId>com.redhat.ba</groupId>
<artifactId>ba-platform-bom</artifactId>
<version>7.13.5.redhat-00002</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
<dependencies>
<!-- Your dependencies -->
</dependencies>
```

2. 在 <dependencies> 标签中声明项目所需的依赖项。将产品 BOM 导入到项目中后，定义了面向用户的依赖项的版本，因此您无需指定这些 <dependency> 元素的 <version> 子元素。但

是，您必须使用 `<dependency>` 元素来声明您要在项目中使用的依赖项。

3.

对于未在 **Business Central** 中编写的独立项目，请指定项目所需的所有依赖项。在您在 **Business Central** 中编写的项目中，**Be Business Central** 会自动提供基本决策引擎和流程引擎依赖项。

-

对于基本的 **Red Hat Process Automation Manager** 项目，根据您要使用的功能，声明以下依赖项：

嵌入式进程引擎依赖项

```

<!-- Public KIE API -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-flow</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-flow-builder</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-bpmn2</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-runtime-manager</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-persistence-jpa</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-query-jpa</artifactId>
</dependency>

```



```

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-audit</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<!-- Dependency needed for default WorkItemHandler implementations. -->
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-workitems-core</artifactId>
</dependency>

<!-- Logging dependency. You can use any logging framework compatible with
slf4j. -->
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>${logback.version}</version>
</dependency>

```

- 对于使用 CDI 的 Red Hat Process Automation Manager 项目，您通常会声明以下依赖项：

启用 CDI 的进程引擎依赖项

```

<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-services-cdi</artifactId>
</dependency>

```

- 对于基本的 Red Hat Process Automation Manager 项目，声明以下依赖项：

嵌入式决策引擎依赖项

```

<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-compiler</artifactId>
</dependency>

<!-- Dependency for persistence support. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-decisiontables</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-templates</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-scorecards</artifactId>
</dependency>

<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
</dependency>

```

- 要使用 KIE 服务器，声明以下依赖项：

客户端应用程序 KIE 服务器依赖项

```

<dependency>
  <groupId>org.kie.server</groupId>

```

```
<artifactId>kie-server-client</artifactId>
</dependency>
```

- 要为 Red Hat Process Automation Manager 创建远程客户端，请声明以下依赖项：

客户端依赖项

```
<dependency>
  <groupId>org.uberfire</groupId>
  <artifactId>uberfire-rest-client</artifactId>
</dependency>
```

- 在创建包含资产的 JAR 文件（如规则和进程定义）时，请将 Maven 项目的打包类型指定为 `kjar`，并使用 `org.kie:kie-maven-plugin` 来处理位于 `<project>` 元素下的 `kjar` 打包类型。在以下示例中，`${kie.version}` 是列出的 Maven 库版本，在 [Red Hat Process Automation Manager 和 Maven 库版本之间有什么映射？](#)

```
<packaging>kjar</packaging>
<build>
  <plugins>
    <plugin>
      <groupId>org.kie</groupId>
      <artifactId>kie-maven-plugin</artifactId>
      <version>${kie.version}</version>
      <extensions>>true</extensions>
    </plugin>
  </plugins>
</build>
```

21.2. 为 BUSINESS CENTRAL 和 KIE 服务器配置外部 MAVEN 存储库

您可以将 Business Central 和 KIE 服务器配置为使用外部 Maven 存储库，如 Nexus 或 Artifactory，而不是内置的存储库。这可以让 Business Central 和 KIE 服务器访问并下载外部 Maven 存储库中维护的工件。



重要

存储库中的工件不会接收自动安全补丁，因为 **Maven** 要求工件不可变。因此，缺少已知安全漏洞的补丁的工件将保留在存储库中，以避免破坏依赖于它们的构建。修补工件的版本号会递增。如需更多信息，请参阅 [JBoss Enterprise Maven Repository](#)。



注意

有关为 **Red Hat OpenShift Container Platform** 上编写环境配置外部 **Maven** 存储库的信息，请参阅以下文档：

- [使用 Operator 在 Red Hat OpenShift Container Platform 4 上部署 Red Hat Process Automation Manager 环境](#)
- [使用模板在 Red Hat OpenShift Container Platform 3 上部署 Red Hat Process Automation Manager 环境](#)

先决条件

- 已安装了 **Business Central** 和 **KIE 服务器**。有关安装选项，请参阅 [规划 Red Hat Process Automation Manager 安装](#)。

流程

1. 创建一个带有连接和访问外部仓库详情的 **Maven settings.xml** 文件。有关 **settings.xml** 文件的详情，请查看 [Maven Settings Reference](#)。
2. 将文件保存到已知位置，例如 `/opt/custom-config/settings.xml`。
3. 在 **Red Hat Process Automation Manager** 安装目录中，导航至 **standalone-full.xml** 文件。例如，如果您使用红帽 **JBoss EAP** 安装红帽流程自动化管理器，则进入 `$EAP_HOME/standalone/configuration/standalone-full.xml`。
4. 打开 **standalone-full.xml**，并在 `<system-properties>` 标签下，将 `kie.maven.settings.custom` 属性设置为 **settings.xml** 文件的完整路径名称。

例如：

```
<property name="kie.maven.settings.custom" value="/opt/custom-config/settings.xml"/>
```

5. 启动或重启 **Business Central** 和 **KIE 服务器**。

后续步骤

对于您要导出或作为 **KJAR** 工件到外部 **Maven** 存储库的每个 **Business Central** 项目，您必须在项目 **pom.xml** 文件中添加存储库信息。具体步骤请查看打包和部署 [Red Hat Process Automation Manager 项目](#)。

21.3. 准备 MAVEN 镜像存储库以离线使用

如果您的 **Red Hat Process Automation Manager** 部署无法对公共互联网进行传出访问，则必须使用所有必要工件的镜像准备 **Maven** 存储库，并使此软件仓库可供您的环境使用。



注意

如果您的 **Red Hat Process Automation Manager** 部署连接到互联网，则不需要完成这个步骤。

先决条件

- 有一个可以传出访问公共互联网的计算机。

流程

1. 在与公共互联网传出连接的计算机上，完成以下步骤：
2. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.13.5
 - a. 下载并解压缩 **Red Hat Process Automation Manager 7.13.5 Offliner Content**

List (rhpam-7.13.5-offliner.zip)产品 deliverable 文件。

- b. 将 rhpam-7.13.5-offliner.zip 文件的内容提取到任何目录中。

- c. 进入该目录并输入以下命令：

```
./offline-repo-builder.sh offliner.txt
```

此命令创建 存储库 子目录，并将必要的工件下载到此子目录中。这是镜像存储库。

如果消息报告一些下载失败，请再次运行相同的命令。如果再次下载失败，请联系红帽支持。

3. 如果您在 **Business Central** 外部开发服务且它们有额外的依赖项，请将依赖项添加到镜像存储库中。如果您将服务开发成 **Maven** 项目，您可以使用以下步骤自动准备这些依赖项。在计算机上完成与公共互联网传出连接的计算机上的步骤。

- a. 创建本地 **Maven** 缓存目录(`~/.m2/repository`)的备份，然后清除该目录。

- b. 使用 `mvn clean install` 命令构建项目的源。

- c. 对于每个项目，输入以下命令来确保 **Maven** 下载项目的所有运行时依赖项：

```
mvn -e -DskipTests dependency:go-offline -f /path/to/project/pom.xml --batch-mode -Djava.net.preferIPv4Stack=true
```

将 `/path/to/project/pom.xml` 替换为项目 `pom.xml` 文件的路径。

- d. 将本地 **Maven** 缓存目录的内容(`~/.m2/ repository`)复制到创建的存储库子目录。

4. 将存储库 子目录的内容复制到您在其上部署 **Red Hat Process Automation Manager** 的计算机的目录中。该目录变为离线 **Maven mirror** 存储库。

5. 为您的 Red Hat Process Automation Manager 部署创建并配置 `settings.xml` 文件，如第 21.2 节“为 Business Central 和 KIE 服务器配置外部 Maven 存储库”所述。

6. 在 `settings.xml` 文件中进行以下更改：

- 在 `<profile>` 标签下，如果缺少 `<repositories>` 或 `<pluginRepositories>` 标签，请添加缺失的标签。

在 `<repositories>` 下添加以下内容：

```
<repository>
  <id>offline-repository</id>
  <url>file:///path/to/repo</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>false</enabled>
  </snapshots>
</repository>
```

使用到本地 Maven mirror repository 目录的完整路径替换 `/path/to/repo`。

- 在 `<pluginRepositories>` 下添加以下内容：

```
<repository>
  <id>offline-plugin-repository</id>
  <url>file:///path/to/repo</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>false</enabled>
  </snapshots>
</repository>
```

使用到本地 Maven mirror repository 目录的完整路径替换 `/path/to/repo`。

7. 将 Business Central 的 `kie.maven.offline.force` 属性设为 `true`。有关设置 Business Central 属性的说明，请参阅在 [Red Hat JBoss EAP 7.4 上安装和配置 Red Hat Process](#)

Automation Manager.

第 22 章 从 GIT 存储库导入项目

Git 是一种分布式版本控制系统。它以提交对象的形式实施修订版本。当您更改保存到存储库时，会在 **Git** 存储库中创建新的提交对象。

Business Central 使用 **Git** 存储项目数据，包括规则和流程等资产。当您在 **Business Central** 中创建项目时，它会被添加到嵌入式 **Business Central** 中的 **Git** 存储库中。如果您在其他 **Git** 存储库中拥有项目，您可以通过 **Business Central** 空间将这些项目导入到 **Business Central Git** 存储库中。

先决条件

- **Red Hat Process Automation Manager** 项目存在于外部 **Git** 存储库中。
- 有对该外部 **Git** 存储库的读取访问权限所需的凭证。

流程

1. 在 **Business Central** 中，点击 **Menu** → **Design** → **Projects**。
2. 选择或创建您要导入项目的空间。默认空间为 **MySpace**。
3. 要导入项目，请执行以下操作之一：
 - 单击 **Import Project**。
 - 从下拉菜单中选择 **Import Project**。
4. 在 **Import Project** 窗口中，输入包含您要导入的项目的 **Git** 存储库的 **URL** 和凭证，然后单击 **Import**。这些项目添加到 **Business Central Git** 存储库，可在当前空间中可用。

第 23 章 自定义 BUSINESS CENTRAL 的品牌

您可以通过用您自己的镜像替换 **Business Central** 登录页面和应用程序标头的品牌。

23.1. 自定义 BUSINESS CENTRAL 登录页面

您可以在 **Business Central** 登录页面中自定义公司徽标和项目徽标。

流程

1. 启动红帽 JBoss EAP 并在网页浏览器中打开 **Business Central**。
2. 将 SVG 格式镜像复制到 Red Hat Process Automation Manager 安装中的 `EAP_HOME/standalone/deployments/business-central.war/img/` 目录中。
3. 在 `EAP_HOME/standalone/deployments/business-central.war/img/` 目录中，可以移动或重命名现有的 `redhat_logo.png` 文件。
4. 重命名您的 PNG 文件 `redhat_logo.png`。
5. 要更改在 **User name** 和 **Password** 字段上方显示的项目徽标，请将默认镜像 `BC_Logo.png` 替换为新的 SVG 文件。
6. 强制完全重新加载登录页面，绕过缓存，以查看更改。例如，在大多数 Linux 和 Windows Web 浏览器中，按 **Ctrl+F5**。

23.2. 自定义 BUSINESS CENTRAL 应用程序标头

您可以自定义 **Business Central** 应用程序标头。

流程

1. 启动红帽 JBoss EAP，在 Web 浏览器中打开 **Business Central**，然后使用您的用户凭据登录。

2. 将 SVG 格式的新应用标头镜像复制到 Red Hat Process Automation Manager 安装中的 `EAP_HOME/standalone/deployments/business-central.war/banner/` 目录中。
3. 在文本编辑器中打开 `EAP_HOME/standalone/deployments/business-central.war/banner/banner.html` 文件。
4. 将 `` 标签中的 `logo.png` 替换为新 `image:admin-and-config/` 的文件名

```

```

5. 强制完全重新加载登录页面，绕过缓存，以查看更改。例如，在大多数 Linux 和 Windows Web 浏览器中，按 `Ctrl+F5`。

第 24 章 集成 LDAP 和 SSL

通过 Red Hat Process Automation Manager, 您可以通过 Red Hat Single Sign-On 集成 LDAP 和 SSL。如需更多信息, 请参阅 [Red Hat Single Sign-On Server 管理指南](#)。

附录 A. BUSINESS CENTRAL 系统属性

本节中列出的 Business Central 系统属性传递到 standalone*.xml 文件。

Git 目录

使用以下属性设置 Business Central Git 目录的位置和名称：

- **org.uberfire.nio.git.dir:** Business Central Git 目录的位置。
- **org.uberfire.nio.git.dirname:** Business Central Git 目录的名称。默认值：`.niogit`。
- **org.uberfire.nio.git.ketch:** Enables 或 disable Git ketch。
- **org.uberfire.nio.git.hooks:** Git hook 目录的位置。

git over HTTP

使用以下属性配置通过 HTTP 对 Git 存储库的访问：

- **org.uberfire.nio.git.proxy.ssh.over.http:** 指定 SSH 是否应使用 HTTP 代理。默认值：`false`。
- **http.proxyHost :** 定义 HTTP 代理的主机名。默认值：`null`。
- **http.proxyPort :** 定义 HTTP 代理的主机端口（整数）。默认值：`null`。
- **http.proxyUser :** 定义 HTTP 代理的用户名。
- **HTTP.proxyPassword :** 定义 HTTP 代理的用户密码。
- **org.uberfire.nio.git.http.enabled:** Enables 或 disable the HTTP 守护进程。默认值：`true`。

- **org.uberfire.nio.git.http.host**: 如果启用了 HTTP 守护进程，它将使用此属性作为主机标识符。这是一个信息性属性，用于显示如何通过 HTTP 访问 Git 存储库。HTTP 仍然依赖于 servlet 容器。默认值：`localhost`。
- **org.uberfire.nio.git.http.hostname**: 如果 HTTP 守护进程被启用，它将使用此属性作为主机名标识符。这是一个信息性属性，用于显示如何通过 HTTP 访问 Git 存储库。HTTP 仍然依赖于 servlet 容器。默认值：`localhost`。
- **org.uberfire.nio.git.http.port**：如果启用了 HTTP 守护进程，它将使用此属性作为端口号。这是一个信息性属性，用于显示如何通过 HTTP 访问 Git 存储库。HTTP 仍然依赖于 servlet 容器。默认值：`8080`。

Git over HTTPS

使用以下属性通过 HTTPS 配置对 Git 存储库的访问：

- **org.uberfire.nio.git.proxy.ssh.over.https**: 指定 SSH 是否使用了 HTTPS 代理。默认值：`false`。
- **HTTPS.proxyHost**：定义 HTTPS 代理的主机名。默认值：`null`。
- **HTTPS.proxyPort**：定义 HTTPS 代理的主机端口（整数值）。默认值：`null`。
- **HTTPS.proxyUser**：定义 HTTPS 代理的用户名。
- **HTTPS.proxyPassword**：定义 HTTPS 代理的用户密码。
- **user.dir**：用户目录的路径。
- **org.uberfire.nio.git.https.enabled**: 启用或禁用 HTTPS 守护进程。默认值：`false`。
- **org.uberfire.nio.git.https.host**: 如果 HTTPS 守护进程已启用，它将使用此属性作为主

机标识符。这是一个信息性属性，用于显示如何通过 HTTPS 访问 Git 存储库。HTTPS 仍然依赖于 servlet 容器。默认值：localhost.

- **org.uberfire.nio.git.https.hostname:** 如果 HTTPS 守护进程已启用，它将使用此属性作为主机名标识符。这是一个信息性属性，用于显示如何通过 HTTPS 访问 Git 存储库。HTTPS 仍然依赖于 servlet 容器。默认值：localhost.

- **org.uberfire.nio.git.https.port:** 如果 HTTPS 守护进程已启用，它将使用此属性作为端口号。这是一个信息性属性，用于显示如何通过 HTTPS 访问 Git 存储库。HTTPS 仍然依赖于 servlet 容器。默认值：8080.

JGit

- **org.uberfire.nio.jgit.cache.instances** : 定义 JGit 缓存大小。
- **org.uberfire.nio.jgit.cache.overflow.cleanup.size:** 定义 JGit 缓存溢出清理大小。
- **org.uberfire.nio.jgit.remove.eldest.iterations:** Enables 或 disable whether remove eldest JGit 迭代。
- **org.uberfire.nio.jgit.cache.evict.threshold.duration:** 定义 JGit 驱除阈值持续时间。
- **org.uberfire.nio.jgit.cache.evict.threshold.time.unit** : 定义 JGit 驱除阈值单元。

Git 守护进程

使用以下属性来启用和配置 Git 守护进程：

- **org.uberfire.nio.git.daemon.enabled:** Enables 或 disable the Git 守护进程。默认值：true.
- **org.uberfire.nio.git.daemon.host:** 如果 Git 守护进程已启用，它将使用此属性作为本地主机标识符。默认值：localhost.
- **org.uberfire.nio.git.daemon.hostname:** 如果 Git 守护进程已启用，它将使用此属性作

为本地主机名标识符。默认值：`localhost`

- **org.uberfire.nio.git.daemon.port:** 如果 Git 守护进程已启用，它将使用此属性作为端口号。默认值：`9418`。
- **org.uberfire.nio.git.http.sslVerify:** 启用或禁用 Git 存储库的 SSL 证书检查。默认值：`true`。



注意

如果已经使用默认或者分配的端口，则会自动选择一个新端口。确保端口可用，并检查日志以了解更多信息。

Git SSH

使用以下属性来启用和配置 Git SSH 守护进程：

- **org.uberfire.nio.git.ssh.enabled:** 启用或禁用 SSH 守护进程。默认值：`true`。
- **org.uberfire.nio.git.ssh.host:** 如果启用了 SSH 守护进程，它将使用此属性作为本地主机标识符。默认值：`localhost`。
- **org.uberfire.nio.git.ssh.hostname:** 如果启用了 SSH 守护进程，它将使用此属性作为本地主机名标识符。默认值：`localhost`。
- **org.uberfire.nio.git.ssh.port:** 如果启用了 SSH 守护进程，它将使用此属性作为端口号。默认值：`8001`。

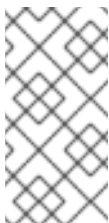


注意

如果已经使用默认或者分配的端口，则会自动选择一个新端口。确保端口可用，并检查日志以了解更多信息。

- **org.uberfire.nio.git.ssh.cert.dir:** 存储本地证书的 `.security` 目录的位置。默认值：`工作目录`。

- **org.uberfire.nio.git.ssh.idle.timeout**:设置 SSH 闲置超时。
- **org.uberfire.nio.git.ssh.passphrase** : 在使用 SCP 风格 URL 克隆 git 存储库时用于访问操作系统的公钥存储的密码短语。示例：`git@github.com:user/repository.git`。
- **org.uberfire.nio.git.ssh.algorithm**: Algorithm 供 SSH 使用。默认值：`RSA`。
- **org.uberfire.nio.git.gc.limit**: Sets the GC 限值。
- **org.uberfire.nio.git.ssh.ciphers** : 以逗号分隔的密码字符串。可用的密码是 `aes128-ctr,aes192-ctr,aes256-ctr,arcfour128,arcfour256,aes192-cbc,aes256-cbc`。如果没有使用 属性，则会载入所有可用的密码。
- **org.uberfire.nio.git.ssh.macs**:以逗号分隔的消息验证代码(MAC)字符串。可用的 MACs 是 `hmac-md5,hmac-md5-96,hmac-sha1,hmac-sha1-96,hmac-sha2-256,hmac-sha2-512`。如果没有使用 属性，则会加载所有可用的 MAC。



注意

如果您计划使用 RSA 或 DSA 以外的任何算法，请确保将应用服务器设置为使用 Bouncy Castle JCE 库。

KIE 服务器节点和流程自动化管理器控制器

使用以下属性，从流程自动化管理器控制器配置与 KIE 服务器节点的连接：

- **org.kie.server.controller**: URL 用于连接到 Process Automation Manager 控制器。例如，`ws://localhost:8080/business-central/websocket/controller`。
- **org.kie.server.user**:用于从流程自动化管理器控制器连接到 KIE 服务器节点的用户名。只有在将这个 Business Central 安装用作流程自动化管理器控制器时才需要此属性。
- **org.kie.server.pwd** : 用于从流程自动化管理器控制器连接到 KIE 服务器节点的

Password。只有在将这个 **Business Central** 安装用作流程自动化管理器控制器时才需要此属性。

Maven 和 miscellaneous

使用以下属性配置 Maven 和其他其它功能：

- **kie.maven.offline.force**：强制 Maven 的行为如离线一样。如果为 **true**，禁用在线依赖关系解析。默认值：**false**。



注意

仅在 **Business Central** 中使用此属性。如果您与任何其他组件共享一个运行时环境，请隔离配置并将其应用到 **Business Central**。

- **org.uberfire.gzip.enable**: Enables 或在 **GzipFilter** 压缩过滤器中禁用 **Gzip** 压缩。默认值：**true**。
- **org.kie.workbench.profile**: 选择 **Business Central** 配置集。可能的值有 **FULL** 或 **PLANNER_AND_RULES**。一个前缀 **FULL_** 设定配置集，并从管理员首选项中隐藏配置集首选项。默认值：**FULL**
- **org.appformer.m2repo.url**: **Business Central** 在查找依赖项时使用 **Maven** 存储库的默认位置。它定向到 **Business Central** 中的 **Maven** 存储库，例如 **http://localhost:8080/business-central/maven2**。在启动 **Business Central** 前设置此属性。默认值：到内 **m2** 存储库的文件路径。
- **appformer.ssh.keystore**：通过指定类名称定义要与 **Business Central** 搭配使用的自定义 **SSH** 密钥存储。如果属性不可用，则使用默认的 **SSH** 密钥存储。
- **appformer.ssh.keys.storage.folder**：在使用默认 **SSH** 密钥存储时，此属性为用户的 **SSH** 公钥定义存储文件夹。如果属性不可用，则密钥将存储在 **Business Central** **.security** 文件夹中。
- **appformer.experimental.features**: 启用实验功能框架。默认值：**false**。

- **org.kie.demo** : 启用来自 GitHub 的演示应用程序的外部克隆。
- **org.uberfire.metadata.index.dir** : 存储 Lucene .index 目录的位置。默认值 : 工作目录。
- **org.uberfire.ldap.regex.role_mapper**: Regex 模式用于将 LDAP 主体名称映射到应用程序角色名称。请注意, 在匹配原则值和角色名称时, 变量角色必须是模式的一部分, 因为应用程序角色名称替换变量角色。
- **org.uberfire.sys.repo.monitor.disabled** : 禁用配置监控器。除非确定, 否则不要禁用。默认值 : false。
- **org.uberfire.secure.key** : 密码加密使用的密码。默认值 : org.uberfire.admin。
- **org.uberfire.secure.alg**: Crypto 算法由密码加密使用。默认值 : PBEWithMD5AndDES。
- **org.uberfire.domain**: uberfire 使用的 Security-domain name。默认值 : ApplicationRealm。
- **org.guvnor.m2repo.dir**: Place 存储 Maven 存储库文件夹。默认值 : < ;working-directory>/repositories/kie。
- **org.guvnor.project.gav.check.disabled**: Disables group ID、工件 ID 和版本(GAV)检查。默认值 : false。
- **org.kie.build.disable-project-explorer** : 禁用 Project Explorer 中所选项目的自动构建。默认值 : false。
- **org.kie.builder.cache.size** : 定义项目构建器的缓存大小。默认值 : 20。
- **org.kie.library.assets_per_page**:您可以在项目屏幕中自定义每个页面的资产数量。默认值 : 15。

- **org.kie.verification.disable-dtable-realtime-verification:** 禁用决策表的实时验证和验证。默认值：**false**。

进程自动化管理器控制器

使用以下属性配置如何连接到流程自动化管理器控制器：

- **org.kie.workbench.controller:** 用于连接 Process Automation Manager 控制器的 URL，例如 `ws://localhost:8080/kie-server-controller/websocket/controller`。
- **org.kie.workbench.controller.user:** Process Automation Manager 控制器用户。默认值：`kieserver`。
- **org.kie.workbench.controller.pwd:** Process Automation Manager 控制器密码。默认值：`kieserver1!`。
- **org.kie.workbench.controller.token:** 用于连接到 Process Automation Manager 控制器的令牌字符串。

Java Cryptography Extension KeyStore(JCEKS)

使用以下属性配置 JCEKS：

- **kie.keystore.keyStoreURL**：用于加载 Java Cryptography Extension KeyStore(JCEKS)的 URL。例如，`file:///home/kie/keystores/keystore.jceks`。
- **kie.keystore.keyStorePwd**：用于 JCEKS 的密码。
- **kie.keystore.key.ctrl.alias:**默认 REST Process Automation Manager 控制器的密钥别名。
- **kie.keystore.key.ctrl.pwd:**默认 REST Process Automation Manager 控制器的别名。

渲染

使用以下属性在 **Business Central** 和 **KIE 服务器** 呈现的形式间切换：

- **org.jbpm.wb.forms.renderer.ext** : 切换 **Business Central** 和 **KIE 服务器** 之间的表单渲染。默认情况下，表单渲染由 **Business Central** 执行。默认值：**false**。
- **org.jbpm.wb.forms.renderer.name**: 让您在 **Business Central** 和 **KIE 服务器** 呈现的表单之间进行切换。默认值：**workbench**。

第 25 章 其他资源

- [在 Red Hat JBoss EAP 集群环境中安装和配置 Red Hat Process Automation Manager](#)

第 26 章 后续步骤

- [将 Red Hat Process Automation Manager 与红帽单点登录集成](#)
- [进程服务的入门](#)

部分 III. 在 RED HAT JBOSS EAP 集群环境中安装和配置 RED HAT PROCESS AUTOMATION MANAGER

作为系统工程师，您可以创建一个 Red Hat Process Automation Manager 集群环境，为您的开发和运行时环境提供高可用性和负载平衡。

先决条件

- 您已查看 [规划 Red Hat Process Automation Manager 安装中的信息](#)。

第 27 章 RED HAT PROCESS AUTOMATION MANAGER 集群

通过集群两个或更多计算机，您可以获得高可用性、增强的协作和负载平衡的好处。当一个计算机出现故障时，高可用性降低了数据丢失的可能性。当计算机出现故障时，另一台计算机通过提供故障计算机上的数据副本来填补差距。当故障计算机再次上线时，它会在集群中恢复它。

您可以通过几种方法集群 Red Hat Process Automation Manager 组件。本文档论述了如何在以下情况下集群：

- [第 28 章 在开发（授权）环境中的 Red Hat Process Automation Manager 集群](#)
- [第 29 章 运行时环境中的 KIE 服务器集群](#)

第 28 章 在开发（授权）环境中的 RED HAT PROCESS AUTOMATION MANAGER 集群

开发人员可以使用 Red Hat Process Automation Manager 编写有助于用户做出决策的规则和流程。

您可以将 Red Hat Process Automation Manager 配置为集群开发环境，以便从高可用性中受益。使用集群环境时，如果开发人员正在处理某个节点且该节点失败，则该开发人员的工作会保留并在集群的任何其他节点上可见。

大多数开发环境由用于创建规则和流程的 Business Central 组成，并且至少是一个 KIE 服务器来测试这些规则和流程。

要创建 Red Hat Process Automation Manager 集群开发环境，您必须执行以下任务：

1.

在作为集群节点的每个系统中配置以下组件：

•

使用 Red Hat Data Grid 8.1 配置 Red Hat JBoss EAP 7.4。

Red Hat Data Grid 由 Infinispan 开源软件项目构建。它是一种分布式内存键/值数据存储，具有索引功能，可让您快速存储、搜索和分析大量数据，近乎实时。在 Red Hat Process Automation Manager 集群环境中，它可让您在集群节点中执行复杂和高效的搜索。

•

配置 AMQ Broker，Java 消息传递服务器(JMS)代理。

JMS 代理是一个软件组件，接收消息，在本地存储它们，然后将消息转发到接收者。AMQ Broker 让您的应用程序与任何消息传递供应商通信。它指定消息驱动的 Bean、企业 JavaBeans 和 servlet 等组件如何发送或接收消息。

•

配置 NFS 文件服务器。

2.

下载 Red Hat JBoss EAP 7.4 和 Red Hat Process Automation Manager 7.13，然后在将成为集群节点的每个系统上安装它们。

3.

在集群的每个节点中配置和启动 Business Central。

28.1. 安装和配置红帽数据网格

要跨集群节点进行更有效的搜索，请安装 Red Hat Data Grid 并针对 Red Hat Process Automation Manager 集群环境进行配置。

有关 Red Hat Data Grid 高级安装和配置选项以及 Red Hat Data Grid 模块的信息，请参阅 [Red Hat Data Grid Server Guide](#)。



注意

不要在与 Business Central 相同的节点上安装 Red Hat Data Grid。

先决条件

- 安装了与 Java 8.0 或更高版本兼容的 Java 虚拟机(JVM)环境。

流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
 - 产品：Data Grid
 - Version: 8.1
2. 下载 Red Hat Data Grid 8.1.0 Server (redhat-datagrid-8.1.0-server.zip)安装文件并将其提取到系统的首选位置。在以下示例中，提取的目录被称为 *JDG_HOME*。
3. 将红帽数据网格更新至最新版本。如需更多信息，请参阅 Red Hat Data Grid [Red Hat Data Grid 用户指南](#)。
4. 要添加 Red Hat Data Grid 用户，请导航至 *JDG_HOME/bin*，并输入以下命令：

```
$ ./cli.sh user create <DATAGRID_USER_NAME> -p <DATA_GRID_PASSWORD> -r default
```

5. 要更改 Red Hat Data Grid 服务器内存参数，打开 `JDG_HOME/bin/server.conf` 文件并找到以下行：

```
-Xms64m -Xmx512m -XX:MetaspaceSize=64M
```

6. 使用以下内容替换此行：

```
-Xms256m -Xmx2048m -XX:MetaspaceSize=256M
```

7. 打开 `JDG_HOME/server/conf/infinispan.xml` 文件并找到以下行：

```
<hotrod-connector name="hotrod"/>
```

8. 使用以下内容替换此行：

```
<hotrod-connector name="hotrod">
  <authentication>
    <sasl mechanisms="SCRAM-SHA-512 SCRAM-SHA-384 SCRAM-SHA-256
      SCRAM-SHA-1 DIGEST-SHA-512 DIGEST-SHA-384
      DIGEST-SHA-256 DIGEST-SHA DIGEST-MD5 PLAIN"
      server-name="infinispan"
      qop="auth"/>
  </authentication>
</hotrod-connector>
```

9. 要运行 Red Hat Data Grid，请导航至 `JDG_HOME` 并输入以下命令：

```
$ ./server.sh -b <HOST>
```

将 `<HOST >` 替换为您安装 Red Hat Data Grid 的服务器的 IP 地址或主机名。

28.2. 下载并配置 AMQ BROKER

Red Hat AMQ Broker 可让您的应用程序与任何消息传递供应商通信。它指定消息驱动的 Bean、企业 JavaBeans 和 servlet 等组件如何发送或接收消息。

有关高级安装和配置选项的详情，请参考 [开始使用 AMQ Broker](#)。

流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：

- **Product: AMQ Broker**

- **版本：7.8.2**

2. 点 Red Hat AMQ Broker 7.8.2 旁边的 **Download (amq-broker-7.8.2-bin.zip)**。

3. 提取 **amq-broker-7.8.2-bin.zip** 文件。

4. 将目录改为 **amq-broker-7.8.2-bin/amq-broker-7.8.2/bin**。

5. 输入以下命令来创建代理和代理用户：

```
./artemis create --host <HOST> --user <AMQ_USER> --password <AMQ_PASSWORD> --require-login <BROKER_NAME>
```

在本例中，替换以下占位符：

- **<HOST >** 是安装 AMQ Broker 的服务器的 IP 地址或主机名。

- **<AMQ_USER >** 和 **<AMQ_PASSWORD >** 是您选择的用户名和密码组合。

- **<BROKER_NAME >** 是您要创建的代理的名称。

6. 要运行 AMQ Broker，请进入 **amq-broker-7.8.2-bin/amq-broker-7.8.2/bin** 目录并输入以下命令：

```
<BROKER_NAME>/bin/artemis run
```

28.3. 配置 NFS 版本 4 服务器

对于一个 **Business Central** 集群环境，需要一个带有共享文件系统的 **NFS 版本 4 服务器**，每个客户端节点都必须有权访问共享文件系统。

流程

1. 配置服务器以导出 **NFS 版本 4 共享**。有关在 **Red Hat Enterprise Linux** 上导出 **NFS 共享** 的步骤，请参考 [管理文件系统](#) 中的 [导出 NFS 共享](#)。有关创建 **NFS 服务器** 的详情，请参考 [如何在 RHEL 7 中配置 NFS](#)。
2. 在服务器中，在文本编辑器中打开 `/etc/exports` 文件。
3. 在 `/etc/exports` 文件中添加以下行，其中 `<HOST_LIST >` 是一个以空格分隔的 **IP 地址列表**，以及被授权连接到服务器的主机的选项：

```
/opt/kie/data <HOST_LIST>
```

例如：

```
/opt/kie/data 192.168.1.0/24(rw,sync) 192.168.1.1/24(no_root_squash)
```

这会创建一个 `/opt/kie/data` 共享，并包含 **NFS** 所需的 `rw,sync,no_root_squash` 最小选项。



注意

您可以使用不同的共享名称而不是 `/opt/kie/data`。如果这样做，您必须在配置运行 **Business Central** 的所有节点时使用不同的名称。

28.4. 下载并提取 RED HAT JBOSS EAP 7.4 和 RED HAT PROCESS AUTOMATION MANAGER

在集群的每个节点上下载并安装 **Red Hat JBoss EAP 7.4** 和 **Red Hat Process Automation Manager 7.13**。

流程

1. 在集群的每个节点中下载 Red Hat JBoss EAP 7.4 :
 - a. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
 - **Product: Enterprise Application Platform**
 - **version: 7.4**
 - b. 点 Red Hat JBoss Enterprise Application Platform 7.4.14 旁边的 Download。 (JBEAP-7.4.14/jboss-eap-7.4.14.zip).
2. 提取 jboss-eap-7.4.14.zip 文件。在以下步骤中，*EAP_HOME* 是 jboss-eap-7.4/jboss-eap-7.4 目录。
3. 下载并应用最新的红帽 JBoss EAP 补丁（如果可用）。
4. 在集群的每个节点中下载 Red Hat Process Automation Manager :
 - a. 进入红帽客户门户网站中的 [Software Downloads](#) 页面，从下拉列表中选择产品和版本：
 - **PRODUCT : 流程自动化管理器**
 - **Version: 7.13.5**
 - b. 下载 Red Hat Process Automation Manager 7.13.5 Business Central Deployable for Red Hat JBoss EAP 7 (rhpam-7.13.5-business-central-eap7-deployable.zip)。
5. 将 rhpam-7.13.5-business-central-eap7-deployable.zip 文件提取到临时目录中。在以下命令中，该目录名为 *TEMP_DIR*。

6. 将 `TEMP_DIR/rhpam-7.13.5-business-central-eap7-deployable/jboss-eap-7.4` 的内容复制到 `EAP_HOME`。
7. 下载并应用最新的 Red Hat Process Automation Manager 补丁（如果可用）。
8. 为您的高可用性环境配置红帽单点登录。如需更多信息，请参阅将 [Red Hat Process Automation Manager](#) 与 [Red Hat Single Sign-On](#) 和 [Red Hat Single Sign-On Server 管理指南](#) 集成。

28.5. 在集群中配置和运行 BUSINESS CENTRAL

安装 Red Hat JBoss EAP 和 Business Central 后，您可以使用 Red Hat Data Grid 和 AMQ Broker 来配置集群。在集群的每个节点中完成以下步骤。



注意

这些步骤描述了基本集群配置。有关更复杂的配置，请参阅 [Red Hat JBoss EAP 7.4 配置指南](#)。



注意

在内部环境的高可用性(HA)中，不要将 KIE 服务器连接到 Business Central。

Business Central 实例无法与每个 KIE 服务器的状态保持同步。例如，如果 KIE 服务器已启动但没有同步，业务中将无法通过该实例进行部署。

先决条件

- Red Hat Data Grid 8.1 已安装如 [第 28.1 节“安装和配置红帽数据网格”](#) 所述。
- 安装并配置 AMQ Broker，如 [第 28.2 节“下载并配置 AMQ Broker”](#) 所述。
- Red Hat JBoss EAP 和 Red Hat Process Automation Manager 安装在集群的每个节点上，如 [第 28.4 节“下载并提取 Red Hat JBoss EAP 7.4 和 Red Hat Process Automation Manager”](#) 所述。

- 有共享文件夹的 NFS 服务器可用，如第 28.3 节“配置 NFS 版本 4 服务器”所述。

流程

1. 要将通过 NFS 共享的目录挂载为 /data，以 root 用户身份输入以下命令：

```
mkdir /data
mount <NFS_SERVER_IP>:<DATA_SHARE> /data
```

将 <NFS_SERVER_IP > 替换为 NFS 服务器系统的 IP 地址或主机名。将 <DATA_SHARE > 替换为您配置的共享名称（例如 /opt/kie/data）。

2. 在 /data NFS 目录中创建一个 kie-wb-playground 目录：

```
mkdir /kie-wb-playground
```

3. 在 *EAP_HOME*/bin 目录中创建一个 kie-wb-playground 目录并挂载该目录：

```
mount -o rw,sync,actimeo=1 <NFS_SERVER_IP>:<DATA_SHARE>/kie-wb-playground kie-wb-playground
```

4. 在文本编辑器中打开 *EAP_HOME*/standalone/configuration/standalone-full.xml 文件。

5. 编辑或添加 < system-properties > 元素中的属性并替换以下占位符：

- <AMQ_USER > 和 <AMQ_PASSWORD > 是您在创建 AMQ Broker 时定义的凭证。
- <AMQ_BROKER_IP_ADDRESS > 是 AMQ Broker 的 IP 地址。
- <DATA_GRID_NODE_IP > 是安装 Red Hat Data Grid 的 IP 地址。
- <SERVER_NAME > 是 Red Hat Data Grid 服务器配置中指定的服务器名称。

- **<SASL_QOP >** 是 Red Hat Data Grid 服务器配置的 **auth**、**auth-int** 和 **auth-conf** 值的组合。
- **<DATAGRID_USER_NAME & gt;** 和 **<DATA_GRID_PASSWORD >** 是您在创建 Red Hat Data Grid 时定义的凭证。

```

<system-properties>
  <property name="appformer-jms-connection-mode" value="REMOTE"/>
  <property name="appformer-jms-username" value="<AMQ_USER>"/>
  <property name="appformer-jms-password" value="<AMQ_USER_PASSWORD>"/>
  <property name="appformer-jms-url"
    value="tcp://<AMQ_BROKER_IP_ADDRESS>:61616?
ha=true&amp;retryInterval=1000&amp;retryIntervalMultiplier=1.0&amp;reconnectAttempts=
-1"/>
  <property name="org.appformer.ext.metadata.infinispan.port"
    value="11222"/>
  <property name="org.appformer.ext.metadata.infinispan.host"
    value="<DATA_GRID_NODE_IP>"/>
  <property name="org.appformer.ext.metadata.infinispan.realm"
    value="ApplicationRealm"/>
  <property name="org.appformer.ext.metadata.infinispan.cluster"
    value="kie-cluster"/>
  <property name="org.appformer.ext.metadata.index"
    value="infinispan"/>
  <property name="org.uberfire.nio.git.dir"
    value="/data"/>
  <property name="es.set.netty.runtime.available.processors"
    value="false"/>
  <property name="org.appformer.ext.metadata.infinispan.username" value="
<DATAGRID_USER_NAME>"/>
  <property name="org.appformer.ext.metadata.infinispan.password" value="
<DATA_GRID_PASSWORD>"/>
  <property name="org.appformer.ext.metadata.index" value="infinispan"/>
  <property name="org.appformer.ext.metadata.infinispan.sasl.qop" value="auth"/>
  <property name="org.appformer.ext.metadata.infinispan.server.name"
value="infinispan"/>
  <property name="org.appformer.ext.metadata.infinispan.realm" value="default"/>
  <property name="org.appformer.concurrent.managed.thread.limit" value="1000"/>
  <property name="org.appformer.concurrent.unmanaged.thread.limit" value="1000"/>
  <property name="org.appformer.concurrent.indexing.thread.limit" value="0"/>
  <property name="org.appformer.ext.metadata.infinispan.server.name"
    value="<SERVER_NAME>"/>
  <property name="org.appformer.ext.metadata.infinispan.sasl.qop"
    value="<SASL_QOP>"/>
</system-properties>

```

6.

保存 **standalone-full.xml** 文件。

7.

要启动集群，请导航到 `EAP_HOME/bin` 并输入以下命令：

```
$ ./standalone.sh -c standalone-full.xml -b <HOST>
```

将 `<HOST >` 替换为安装 Red Hat Process Automation Manager 的服务器的 IP 地址或主机名。

28.6. 测试您的高可用性(HA)内部基础架构

当您为 **Business Central** 创建生产就绪的高可用性(HA)内部基础架构时，您必须确保它满足可行 HA 环境的最小硬件和性能要求。HA 内部基础架构由以下四个主要组件组成：业务中心、消息系统(AMQ)、索引服务器(Red Hat Data Grid)和共享文件系统(NFS/GlusterFS/Ceph)。

先决条件

•

至少 3 个节点的网络环境使用以下布局：

节点 1：Business Central

节点 2：Business Central

节点 3：AMQ、Red Hat Data Grid 和 NFS

流程

1.

测试网络速度：

a.

在每台服务器节点的命令终端中，安装 **iPerf3**：

```
$ dnf install iperf3
```

b.

在 **NFS 服务器节点（服务器节点 3）** 的命令终端中，在服务器模式中启动 **iPerf3**：

```
$ iperf3 -s
```

c.

在每个 **Business Central** 服务器节点的命令终端中，输入以下命令在客户端模式中启动 **iPerf3**，并将 **NFS** 服务器节点设置为主机：

```
$ iperf3 -c <NFS_SERVER_IP>
```

+

In this example, replace ``<NFS_SERVER_IP>`` with the IP address of the NFS server.

d.

将每个服务器节点的结果与以下最小值的示例进行比较：

```
iperf3 -c 172.31.47.103
Connecting to host 172.31.47.103, port 5201
[ 5] local 172.31.39.4 port 44820 connected to 172.31.47.103 port 5201
[ ID] Interval      Transfer  Bitrate   Retr  Cwnd
[ 5] 0.00-1.00 sec  143 MBytes 1.20 Gbits/sec  0  419 KBytes
[ 5] 1.00-2.00 sec  111 MBytes 928 Mbits/sec  6  848 KBytes
[ 5] 2.00-3.00 sec  53.8 MBytes 451 Mbits/sec  0  1.08 MBytes
[ 5] 3.00-4.00 sec  52.5 MBytes 440 Mbits/sec  1  1022 KBytes
[ 5] 4.00-5.00 sec  53.8 MBytes 451 Mbits/sec  1  935 KBytes
[ 5] 5.00-6.00 sec  53.8 MBytes 451 Mbits/sec  1  848 KBytes
[ 5] 6.00-7.00 sec  52.5 MBytes 440 Mbits/sec  0  1.08 MBytes
[ 5] 7.00-8.00 sec  53.8 MBytes 451 Mbits/sec  1  1.01 MBytes
[ 5] 8.00-9.00 sec  53.8 MBytes 451 Mbits/sec  1  953 KBytes
[ 5] 9.00-10.00 sec 52.5 MBytes 440 Mbits/sec  1  856 KBytes
-----
[ ID] Interval      Transfer  Bitrate   Retr
[ 5] 0.00-10.00 sec  680 MBytes 570 Mbits/sec  12      sender
[ 5] 0.00-10.04 sec  677 MBytes 566 Mbits/sec           receiver

iperf Done.
```

2.

验证 **NFS** 信息：

a.

在每个 **Business Central** 服务器节点的命令终端中挂载 **NFS** 节点：

```
$ mount -o actimeo=1 <NFS_SERVER_IP>:/opt/nfs/kie /opt/kie/niogit
```

b.

在每个挂载的节点的命令终端中，输入 **nfsiostat**：

```
$ nfsiostat
```

c.

将每个服务器节点的结果与以下最小值的示例进行比较：

```

nfsiostat
ops/s rpc bklog
6.415 0.000

read:
ops/s kB/s kB/op retrans avg RTT (ms) avg exe (ms) avg queue (ms) errors
0.031 0.045 1.452 0 (0.0%) 0.129 0.166 0.019 0 (0.0%)

write:
ops/s kB/s kB/op retrans avg RTT (ms) avg exe (ms) avg queue (ms) errors
0.517 0.467 0.903 0 (0.0%) 1.235 1.269 0.018 0 (0.0%)

```

3.

验证磁盘是否是一个 SSD :

a.

在 NFS 服务器的命令终端中，输入 `df -h` 来识别磁盘，如下例所示：

```

$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G  0 3.8G  0% /dev
tmpfs           3.9G  0 3.9G  0% /dev/shm
tmpfs           3.9G  33M 3.8G  1% /run
tmpfs           3.9G  0 3.9G  0% /sys/fs/cgroup
/dev/xvda2      25G  3.2G 22G 13% /
tmpfs           781M  0 781M  0% /run/user/1000
172.31.47.103:/root/nfs 25G  2.1G 23G  9% /root/nfs

```

b.

输入 `lsblk -d` 验证磁盘是否是一个 SSD :

```

$ lsblk -d

```

c.

输入 `hdparm -Tt` 来测试磁盘：

```

$ hdparm -Tt /dev/xvda2

```

d.

将每个服务器节点的结果与以下最小值的示例进行比较：

```

$ hdparm -Tt /dev/xvda2

/dev/xvda2:
Timing cached reads: 18670 MB in 1.99 seconds = 9389.01 MB/sec
Timing buffered disk reads: 216 MB in 3.03 seconds = 71.40 MB/sec

```

为 Red Hat Process Automation Manager 配置集群后，创建一个资产来验证安装是否正常工作。

流程

1. 在网页浏览器中，输入 `<node-IP-address>:8080/business-central`。将 `<node-IP-address >` 替换为特定节点的 IP 地址。
2. 输入您在安装过程中创建的 **admin** 用户凭证。此时会出现 **Business Central** 主页。
3. 选择 **Menu** → **Design** → **Projects**。
4. 打开 **MySpace** 空间。
5. 点 **Try Samples** → **Mortgages** → **OK**。此时会出现 **Assets** 窗口。
6. 点 **Add Asset** → **Data Object**。
7. 在 **Data Object** 字段中输入 **MyDataObject**，然后单击 **OK**。
8. 点 **Spaces** → **MySpace** → **Mortgages** 并确认 **MyDataObject** 位于资产列表中。
9. 在网页浏览器中输入以下 URL，其中 `& lt;node_IP_address >` 是集群不同节点的地址：

`http://<node_IP_address>:8080/business-central`
10. 输入您用于在第一个节点上登录 **Business Central** 的同一凭证，在其中创建了 **MyDataObject** 资产。
11. 选择 **Menu** → **Design** → **Projects**。

12. 打开 **MySpace** 空间。
13. 选择 **Mortgages** 项目。
14. 验证 **MyDataObject** 是否在 **asset** 列表中。
15. 删除 **Mortgages** 项目。

第 29 章 运行时环境中的 KIE 服务器集群

在运行时环境中，KIE 服务器运行包含支持业务决策的规则和流程的服务。集群 KIE 服务器运行时环境的主要优点是负载均衡。如果集群的一个节点中的活动增加，则该活动可以在集群的其他节点间共享，以提高性能。

要创建 KIE 服务器集群运行时环境，您可以下载并提取 Red Hat JBoss EAP 7.4 和 KIE 服务器。然后，您可以为集群配置红帽 JBoss EAP 7.4，启动集群，并在每个集群节点上安装 KIE 服务器。

另外，您还可以将无头进程自动化管理器控制器和智能路由器集群。



注意

在内部环境的高可用性(HA)中，不要将 KIE 服务器连接到 Business Central。

Business Central 实例无法与每个 KIE 服务器的状态保持同步。例如，如果 KIE 服务器已启动但没有同步，业务中将无法通过该实例进行部署。

29.1. 下载并提取红帽 JBOSS EAP 7.4 和 KIE 服务器

完成本节中的步骤，下载并安装用于在集群环境中安装的 Red Hat JBoss EAP 7.4 和 KIE 服务器。

流程

1. 在集群的每个节点中下载 Red Hat JBoss EAP 7.4 :
 - a. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
 - **Product: Enterprise Application Platform**
 - **version: 7.4**
 - b. 单击 Red Hat JBoss Enterprise Application Platform 7.4.14 旁边的 Download。

(jboss-eap-7.4.14.zip)。

2. 提取 jboss-eap-7.4.14.zip 文件。jboss-eap-7.4/jboss-eap-7.4 目录称为 *EAP_HOME*。
3. 下载并应用最新的红帽 JBoss EAP 补丁（如果可用）。
4. 下载 KIE 服务器：
 - a. 进入红帽客户门户网站中的 [Software Downloads](#) 页面，从下拉列表中选择产品和版本：
 - **PRODUCT**：流程自动化管理器
 - **Version**: 7.13.5
 - b. 下载所有支持的 EE8 容器的 Red Hat Process Automation Manager 7.13.5 KIE Server (rhpm-7.13.5-kie-server-ee8.zip)。
5. 将 rhpm-7.13.5-kie-server-ee8.zip 存档提取到临时目录中。在以下示例中，此目录名为 *TEMP_DIR*。
6. 将 *TEMP_DIR*/rhpm-7.13.5-kie-server-ee8/rhpm-7.13.5-kie-server-ee8/kie-server.war 目录复制到 *EAP_HOME*/standalone/deployments/。



警告

确定您复制的 Red Hat Process Automation Manager 部署的名称不会与您在 Red Hat JBoss EAP 实例中的现有部署冲突。

7. 将 *TEMP_DIR*/rhpm-7.13.5-kie-server-ee8/rhpm-7.13.5-kie-server-

ee8/SecurityPolicy/ 的内容复制到 *EAP_HOME/bin*。

8. 当系统提示覆盖文件时，点 替换。
9. 在 *EAP_HOME/standalone/deployments/* 目录中，创建名为 *kie-server.war.dodeploy* 的空文件。此文件确保服务器启动时会自动部署 KIE 服务器。
10. 下载并应用最新的 Red Hat Process Automation Manager 补丁（如果可用）。
11. 导航到 *EAP_HOME/bin* 目录。
12. 使用 *kie-server* 角色创建一个用户，用于登录到 KIE 服务器。

```
$. /bin/jboss-cli.sh --commands="embed-server --std-out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>),/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>'}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=['kie-server'])"
```

13. 记录您的用户名和密码。

29.2. 为 KIE 服务器配置并运行 RED HAT JBOSS EAP 7.4 集群

为 KIE 服务器配置红帽 JBoss EAP 集群，然后启动该集群。

流程

1. 在所有属于此集群的红帽 JBoss EAP 实例上安装 JDBC 驱动程序。如需更多信息，请参阅 *Red Hat JBoss EAP 7.4 配置指南* 中的 "[JDBC 驱动程序](#)" 部分。
2. 在文本编辑器中打开 *EAP_HOME/standalone/configuration/standalone-full.xml* 文件。

3.

编辑 `data-stores` 属性以及其上面的 `timer-service thread-pool-name` :

- `datasource-jndi-name` 是此流程开始时指定的数据库的 JNDI 名称。
- 您可以为 `partition` 属性的值输入任何名称。但是，节点只会看到来自具有相同分区名称的其他节点的计时器。通过分配分区名称将节点分组到分区中，您可以将大型集群分成几个较小的集群。这样做可以提高性能。例如，某个集群不是 100 个节点，其中所有 100 个节点都试图执行和刷新相同的计时器，您可以通过为每个组 5 个不同的分区名称来创建 20 个集群。
- 将 `default-data-store` 属性值替换为 `ejb_timer_ds`。
- 以毫秒为单位设置 `refresh-interval` 值，以指定 EJB 计时器连接到数据库以同步和加载任务的频率。

```
<timer-service thread-pool-name="default" default-data-store="ejb_timer_ds">
<data-stores>
  <database-data-store name="ejb_timer_ds" datasource-jndi-
name="java:jboss/datasources/ejb_timer" database="postgresql"
partition="ejb_timer_part" refresh-interval="30000"/>
</data-stores>
</timer-service>
```

下表列出了支持的数据库以及对应的数据库属性值：

表 29.1. 支持的数据库

数据库	属性值
Hyper SQL (仅用于演示目的, 不支持)	hsql
PostgreSQL	postgresql
Oracle	oracle
IBM DB2	db2
Microsoft SQL Server	mssql
MySQL 和 MariaDB	mysql

4.

将 KIE 服务器和 EJB 计时器数据源添加到 `standalone-full.xml` 文件。在这些示例中，`<DATABASE>` 是数据库的名称，`<SERVER_NAME>` 是 JNDI 数据库的主机名，`<USER_NAME>` 和 `<USER_PWD>` 是该数据库的凭据。

- 添加数据源以允许 KIE 服务器连接到数据库，例如：

```
<xa-datasource jndi-name="java:/jboss/datasources/rhpam" pool-name="rhpam-RHPAM"
use-java-context="true" enabled="true">
  <xa-datasource-property name="DatabaseName"><DATABASE></xa-datasource-
property>
  <xa-datasource-property name="PortNumber">5432</xa-datasource-property>
  <xa-datasource-property name="ServerName"><SERVER_NAME></xa-datasource-
property>
  <driver>postgresql</driver>
  <security>
    <user-name><USER_NAME></user-name>
    <password><USER_PWD></password>
  </security>
</xa-datasource>
```

- 添加数据源以启用 EJB 计时器，例如：

```
<xa-datasource jndi-name="java:/jboss/datasources/ejb_timer" pool-name="ejb_timer"
use-java-context="true" enabled="true">
  <xa-datasource-property name="DatabaseName"><DATABASE></xa-datasource-
property>
  <xa-datasource-property name="PortNumber">5432</xa-datasource-property>
  <xa-datasource-property name="ServerName"><SERVER_NAME></xa-datasource-
property>
  <driver>postgresql</driver>
  <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
  <security>
    <user-name><USER_NAME></user-name>
    <password><USER_PWD></password>
  </security>
</xa-datasource>
```



警告

您必须对 KIE 服务器运行时数据和 EJB 计时器数据使用两个不同的数据库。

5.

在 `<system-properties>` 元素中添加以下属性，并替换以下占位符：

- `<JNDI_NAME>` 是数据源的 JNDI 名称。对于 Red Hat Process Automation Manager，这是 `java:/jboss/datasources/rhpam`。
- `<DIALECT>` 是您数据库的休眠状态。

支持以下分区：

- **DB2:** `org.hibernate.dialect.DB2Dialect`
- **MSSQL:** `org.hibernate.SQLServer 2012Dialect`
- **MySQL:** `org.hibernate.dialect.MySQL5InnoDBDialect`
- **MariaDB :** `org.hibernate.dialect.MySQL5InnoDBDialect`
- **Oracle :** `org.hibernate.dialect.Oracle10gDialect`
- **PostgreSQL :** `org.hibernate.dialect.PostgreSQL82Dialect`
- **PostgreSQL 加上 :** `org.hibernate.dialect.PostgresPlusPlusDialect`

```
<system-properties>
  <property name="org.kie.server.persistence.ds" value="<JNDI_NAME>"/>
  <property name="org.kie.server.persistence.dialect" value="<DIALECT>"/>
  <property name="org.jbpm.ejb.timer.tx" value="true"/>
</system-properties>
```

6.

保存 `standalone-full.xml` 文件。

7.

要启动集群，请导航到 *EAP_HOME/bin* 并输入以下命令之一：

- 在 Linux 或基于 UNIX 的系统中：

```
$./standalone.sh -c standalone-full.xml
```

- 在 Windows 中：

```
standalone.bat -c standalone-full.xml
```

29.3. 使用无头进程自动化管理器控制器集群 KIE 服务器

流程自动化管理器控制器与 Business Central 集成。但是，如果您不安装 Business Central，则可安装无头流程自动化管理器控制器，并使用 REST API 或 KIE 服务器 Java 客户端 API 与它交互。

先决条件

- 已备份的红帽 JBoss EAP 安装版本 7.4 或更高版本。红帽 JBoss EAP 安装的基础目录称为 *EAP_HOME*。
- 需要足够的用户权限以完成安装。
- 有共享文件夹的 NFS 服务器可用，如第 28.3 节“配置 NFS 版本 4 服务器”所述。

流程

1.

进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：

- **PRODUCT**：流程自动化管理器

- **Version**: 7.13.5

2.

下载 Red Hat Process Automation Manager 7.13.5 Add Ons ([rhpam-7.13.5-add-](#)

ons.zip 文件)。

3. 提取 `rhpm-7.13.5-add-ons.zip` 文件。`rhpm-7.13.5-controller-ee7.zip` 文件位于提取的目录中。
4. 将 `rhpm-7.13.5-controller-ee7.zip` 存档提取到临时目录中。在以下示例中，此目录名为 `TEMP_DIR`。
5. 将 `TEMP_DIR/rhpm-7.13.5-controller-ee7/controller.war` 目录复制到 `EAP_HOME/standalone/deployments/`。



警告

确保您复制的无头进程自动化管理器控制器部署的名称与您在 Red Hat JBoss EAP 实例中的现有部署不冲突。

6. 将 `TEMP_DIR/rhpm-7.13.5-controller-ee7/SecurityPolicy/` 目录的内容复制到 `EAP_HOME/bin`。
7. 当系统提示覆盖文件时，请单击 **Yes**。
8. 在 `EAP_HOME/standalone/deployments/` 目录中，创建名为 `controller.war.dodeploy` 的空文件。此文件确保服务器启动时自动部署无头进程自动化管理器控制器。
9. 在文本编辑器中打开 `EAP_HOME/standalone/configuration/standalone.xml` 文件。
10. 在 `<system-properties>` 元素中添加以下属性，并将 `< ;NFS_STORAGE >` 替换为存储模板配置的 NFS 存储的绝对路径：

```
<system-properties>
  <property name="org.kie.server.controller.templatefile.watcher.enabled" value="true"/>
  <property name="org.kie.server.controller.templatefile" value="<NFS_STORAGE>"/>
</system-properties>
```

```
</system-properties>
```

模板文件包含特定部署场景的默认配置。

如果将 `org.kie.server.controller.templatefile.watcher.enabled` 属性的值设置为 `true`，则会启动一个单独的线程来监视模板文件的修改。这些检查的默认间隔为 30000 毫秒，可以由 `org.kie.server.controller.templatefile.watcher.interval` 系统属性进一步控制。如果此属性的值设为 `false`，则仅在服务器重启时检测到对模板文件的更改。

11.

要启动无头进程自动化管理器控制器，请导航到 `EAP_HOME/bin`，然后输入以下命令：

- 在 Linux 或基于 UNIX 的系统中：

```
$./standalone.sh
```

- 在 Windows 中：

```
standalone.bat
```

29.4. 在失败的 KIE 服务器节点上为 RUNNING 状态配置作业故障切换

默认情况下，当在一个 KIE 服务器节点中运行多个 KIE 服务器节点时，如果一个 KIE 服务器节点失败，其他可用的 KIE 服务器节点会获取 `QUEUED` 或 `RETRYING` 状态来自失败的 KIE 服务器节点。但是，这些节点不会获取处于 `RUNNING` 状态的作业。

您可以将集群配置为在 `RUNNING` 状态中包含作业，以便重新排队到另一个 KIE 服务器节点。如果一个计时器在失败的 KIE 服务器节点上的进程中启动节点，则该进程仅在其中一个节点上启动。

先决条件

- 您已为 KIE 服务器配置了红帽 JBoss EAP 集群。

流程

1. 准备环境：

a.

验证 `standalone.xml` 文件中是否存在以下扩展，并根据需要添加它们：

```
<extension module="org.jboss.as.clustering.infinispan"/>
<extension module="org.jboss.as.clustering.jgroups"/>
```

b.

在 `standalone.xml` 文件中，找到 `<subsystem xmlns="urn:jboss:domain:infinispan:9.0">` 子系统，并创建一个名为 `jbpm` 的缓存容器条目，其中包含一个名为 `nodes` 的缓存，如下例所示：

```
<cache-container name="jbpm">
  <transport lock-timeout="60000"/>
  <replicated-cache name="nodes">
    <transaction mode="BATCH"/>
  </replicated-cache>
  <replicated-cache name="jobs">
    <transaction mode="BATCH"/>
  </replicated-cache>
</cache-container>
```

c.

为复制和分布式缓存添加 `jgroups` 子系统：

```
<subsystem xmlns="urn:jboss:domain:jgroups:7.0">
  <channels default="ee">
    <channel name="ee" stack="udp" cluster="ejb"/>
  </channels>
  <stacks>
    <stack name="udp">
      <transport type="UDP" socket-binding="jgroups-udp"/>
      <protocol type="PING"/>
      <protocol type="MERGE3"/>
      <socket-protocol type="FD_SOCKET" socket-binding="jgroups-udp-fd"/>
      <protocol type="FD_ALL"/>
      <protocol type="VERIFY_SUSPECT"/>
      <protocol type="pbcast.NAKACK2"/>
      <protocol type="UNICAST3"/>
      <protocol type="pbcast.STABLE"/>
      <protocol type="pbcast.GMS"/>
      <protocol type="UFC"/>
      <protocol type="MFC"/>
      <protocol type="FRAG3"/>
    </stack>
    <stack name="tcp">
      <transport type="TCP" socket-binding="jgroups-tcp"/>
      <socket-protocol type="MPING" socket-binding="jgroups-mping"/>
      <protocol type="MERGE3"/>
      <socket-protocol type="FD_SOCKET" socket-binding="jgroups-tcp-fd"/>
      <protocol type="FD_ALL"/>
      <protocol type="VERIFY_SUSPECT"/>
      <protocol type="pbcast.NAKACK2"/>
```

```

<protocol type="UNICAST3"/>
<protocol type="pbcast.STABLE"/>
<protocol type="pbcast.GMS"/>
<protocol type="MFC"/>
<protocol type="FRAG3"/>
</stack>
</stacks>
</subsystem>

```

d.

配置以下接口：

```

<interface name="private">
  <inet-address value="{jboss.bind.address.private:127.0.0.1}"/>
</interface>

```

e.

配置以下套接字：

```

<socket-binding name="jgroups-mping" interface="private" multicast-
address="{jboss.default.multicast.address:230.0.0.4}" multicast-port="45700"/>

<socket-binding name="jgroups-tcp" interface="private" port="7600"/>
<socket-binding name="jgroups-tcp-fd" interface="private" port="57600"/>
<socket-binding name="jgroups-udp" interface="private" port="55200" multicast-
address="{jboss.default.multicast.address:230.0.0.4}" multicast-port="45688"/>
<socket-binding name="jgroups-udp-fd" interface="private" port="54200"/>

```

2.

将 Infinispan 扩展添加到 KIE 服务器：

a.

在文本编辑器中打开 `kie-server.war/WEB-INF/jboss-deployment-structure.xml` 文件，并在文件中添加以下依赖项：

```

<module name="org.infinispan" services="export"/>
<module name="org.jgroups" />

```

b.

下载 JAR 文件。

i.

进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：

- 产品：流程自动化管理器

- - Version: 7.13.5
- ii. 下载 Red Hat Process Automation Manager 7.13.5 Maven 存储库 (rhpm-7.13.5-maven-repository.zip)并提取 ZIP 文件。
- iii. 导航到包含提取的文件的目录，并将 maven-repository/org/kie/server/kie-server-services-jbpm-cluster/7.67.0.Final-redhat-00024/kie-server-services-jbpm-cluster-7.67.0.Final-redhat-00024.jar 文件复制到 kie-server.war/WEB-INF/lib 目录中。

现在，集群被配置为在 **RUNNING** 状态下为作业使用故障转移。

第 30 章 安装和配置智能路由器

智能路由器（KIE 服务器路由器）是一个轻量级 Java 组件，您可用作多个 KIE 服务器、客户端应用程序和其他组件之间的集成层。根据您的部署和执行环境，智能路由器可以聚合多个独立的 KIE 服务器实例，就像它们是单一服务器一样。智能路由器提供以下功能：

数据聚合

当有客户端应用程序请求并聚合结果时，从所有 KIE 服务器实例（一个实例）收集数据。

路由

作为单一端点，接收从客户端应用程序调用到任何服务，并将每个调用自动路由到运行特定服务的 KIE 服务器。这意味着 KIE 服务器不需要部署相同的服务。

负载均衡

提供有效的负载均衡。智能路由器集群的负载均衡请求必须使用标准负载均衡工具在外部管理。

身份验证

使用系统属性标记来验证 KIE 服务器实例，并可以启用 HTTPS 流量。

环境管理

管理更改环境，例如添加或删除服务器实例。

30.1. 使用智能路由器负载均衡 KIE 服务器实例

您可以使用智能路由器聚合多个独立的 KIE 服务器实例，就像它们是单一服务器一样。它执行智能负载均衡器的角色，因为它可以将请求路由到单独的 KIE 服务器实例，并聚合来自不同 KIE 服务器实例的数据。智能路由器使用别名作为代理来执行。

先决条件

- 安装了多个 KIE 服务器实例。



注意

您不需要将 KIE 服务器配置为智能路由器的非受管。

非受管 KIE 服务器实例不会连接到控制器。例如，如果您将非受管 KIE 服务器实例连接到智能路由器，并使用控制器注册智能路由器，那么 Business Central 通过使用智能路由器联系非受管 KIE 服务器实例。

流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
 - **PRODUCT**：流程自动化管理器
 - **Version**: 7.13.5
2. 下载 Red Hat Process Automation Manager 7.13.5 附加组件。
3. 将下载的 `rhpm-7.13.5-add-ons.zip` 文件提取到临时目录中。`rhpm-7.13.5-smart-router.jar` 文件位于提取的 `rhpm-7.13.5-add-ons` 目录中。
4. 将 `rhpm-7.13.5-smart-router.jar` 文件复制到您要运行该文件的位置。
5. 输入以下命令启动智能路由器：

```
java
-Dorg.kie.server.router.host=<ROUTER_HOST>
-Dorg.kie.server.router.port=<ROUTER_PORT>
-Dorg.kie.server.controller=<CONTROLLER_URL>
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.router.config.watcher.enabled=true
-Dorg.kie.server.router.repo=<NFS_STORAGE>
-jar rhpm-7.13.5-smart-router.jar
```

以上命令中的属性具有以下默认值：

```
org.kie.server.router.host=localhost
org.kie.server.router.port=9000
org.kie.server.controller= N/A
org.kie.server.controller.user=kieserver
org.kie.server.controller.pwd=kieserver1!
org.kie.server.router.repo= <CURRENT_WORKING_DIR>
org.kie.server.router.config.watcher.enabled=false
```

`org.kie.server.controller` 是服务器控制器的 URL，例如：

```
org.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller
```

`org.kie.server.router.config.watcher.enabled` 是一个可选设置，用于启用 `watcher` 服务系统属性。



注意

您可以使用配置文件，而不必在命令行中指定配置属性。有关使用文件配置智能路由器的详情，请参考 [第 30.5 节“使用配置文件配置智能路由器设置”](#)。

6. 在必须连接到智能路由器的每个 KIE 服务器实例上，将 `org.kie.server.router` 系统属性设置为 **Smart Router URL**。

7. 要从客户端访问智能路由器，请使用智能路由器 URL 而不是 KIE 服务器 URL，例如：

```
KieServicesConfiguration config =
KieServicesFactory.newRestConfiguration("http://smartrouter.example.com:9000",
"USERNAME", "PASSWORD");
```

在本例中，`smartrouter.example.com` 是 Smart Router URL，`USERNAME` 和 `PASSWORD` 是智能路由器配置的凭据。

8. 要在 `unmanaged` KIE 服务器中创建新容器，以便您可以使用示例数据进行填充，请发送以下 HTTP 请求：

```
$ curl -v -X POST -H 'Content-type: application/xml' -H 'X-KIE-Content-Type: xstream' -d
@create-container.xml -u ${KIE_CRED} http://${KIE_SERVER_HOST}:${KIE_SERVER-
PORT}/kie-server/services/rest/server/config/
```

9.

查看 **create-container.xml** 文件的内容：

```
<script>
<create-container>
  <container container-id="example:timer-test:1.1">
    <release-id>
      <group-id>example</group-id>
      <artifact-id>timer-test</artifact-id>
      <version>1.1</version>
    </release-id>
    <config-items>
      <itemName>RuntimeStrategy</itemName>
      <itemValue>PER_PROCESS_INSTANCE</itemValue>
      <itemType></itemType>
    </config-items>
  </container>
</create-container>
</script>
```

在 **Smart Router** 控制台中会显示有关部署容器的消息。例如：

```
INFO: Added http://localhost:8180/kie-server/services/rest/server as server location for
container example:timer-test:1.1
```

10.

要显示容器列表，请输入以下命令：

```
$ curl http://localhost:9000/mgmt/list
```

此时会显示容器列表：

```
{
  "containerInfo": [{
    "alias": "timer-test",
    "containerId": "example:timer-test:1.1",
    "releaseId": "example:timer-test:1.1"
  }],
  "containers": [
    {"example:timer-test:1.1": ["http://localhost:8180/kie-server/services/rest/server"]},
    {"timer-test": ["http://localhost:8180/kie-server/services/rest/server"]}
  ],
  "servers": [
    {"kieserver2": []},
```

```

    {"kieserver1": ["http://localhost:8180/kie-server/services/rest/server"]}
  ]
}

```

11.

要使用智能路由器 URL 启动进程，请输入以下命令：

```

$ curl -s -X POST -H 'Content-type: application/json' -H 'X-KIE-Content-Type: json' -d
'{"timerDuration": "9s"}' -u kieserver:kieserver1!
http://localhost:9000/containers/example:timer-test:1.1/processes/timer-
test.TimerProcess/instances

```

30.2. 为 TLS 支持配置智能路由器

您可以为传输层安全(TLS)支持配置智能路由器（KIE 服务器路由器）来允许 HTTPS 流量。另外，您可以禁用未安全 HTTP 连接至智能路由器。

先决条件

- KIE 服务器安装在红帽 JBoss EAP 7.4 集群的每个节点中。
- 已安装并配置了智能路由器。更多信息请参阅 [第 30.1 节“使用智能路由器负载均衡 KIE 服务器实例”](#)。

流程

要启动智能路由器，请使用以下方法之一：

- 要启动带有 TLS 支持和启用 HTTPS 的智能路由器以及允许 HTTP 连接，请输入以下命令：

```

java -Dorg.kie.server.router.tls.keystore = <KEYSTORE_PATH>
-Dorg.kie.server.router.tls.keystore.password = <KEYSTORE_PASSWORD>
-Dorg.kie.server.router.tls.keystore.keyalias = <KEYSTORE_ALIAS>
-Dorg.kie.server.router.tls.port = <HTTPS_PORT>
-jar rhpam-7.13.5-smart-router.jar

```

在本例中，替换以下变量：

- <KEYSTORE_PATH >：将要存储密钥存储的路径。

- `<KEYSTORE_PASSWORD>` : 密钥存储密码。
- `<KEYSTORE_ALIAS >` : 用于存储证书的别名。
- `<HTTPS_PORT>` : HTTPS 端口。默认 HTTPS 端口为 9443。
- 要启动带有 TLS 支持和启用了 HTTPS 的智能卡并禁用 HTTP 连接，请输入以下命令：

```
java -Dorg.kie.server.router.tls.keystore = <KEYSTORE_PATH>
-Dorg.kie.server.router.tls.keystore.password = <KEYSTORE_PASSWORD>
-Dorg.kie.server.router.tls.keystore.keyalias = <KEYSTORE_ALIAS>
-Dorg.kie.server.router.tls.port = <HTTPS_PORT>
-Dorg.kie.server.router.port=0
-jar rhpam-7.13.5-smart-router.jar
```

当 `org.kie.server.router.port` 系统属性设置为 0 时，HTTP 侦听器不会被注册。如果配置了 TLS 且 HTTP 侦听器没有注册，则智能路由器仅侦听 HTTPS 端口。



注意

如果没有配置 TLS，并且您可以通过将 `org.kie.server.router.port` 设置为 0 来禁用 HTTP，则会出现错误并智能路由器停止。

30.3. 为端点身份验证配置智能路由器

您可以配置智能路由器（KIE 服务器路由器）进行端点身份验证。

先决条件

- KIE 服务器安装在红帽 JBoss EAP 7.4 集群的每个节点中。
- 已安装并配置了智能路由器。更多信息请参阅 [第 30.1 节“使用智能路由器负载均衡 KIE 服务器实例”](#)。

流程

- 要使用启用端点身份验证来启动智能路由器，请配置管理凭证：

- a. 在您的 KIE 服务器配置中添加以下属性：

```
`org.kie.server.router.management.username`  
`org.kie.server.router.management.password`
```

默认用户名是 KIE 服务器 ID。

- b. 在您的智能路由器配置中添加以下属性：

```
`org.kie.server.router.management.password`
```

密码属性值为 **true** 或 **false**（默认值）。



注意

启用端点身份验证意味着任何列出、添加或删除容器的操作都必须进行身份验证。

1. 可选：将用户添加到智能路由器。例如：

```
java -jar rhpam-7.13.5-smart-router.jar -addUser <USERNAME> <PASSWORD>
```

2. 可选：从智能路由器中删除用户。例如：

```
java -jar rhpam-7.13.5-smart-router.jar -removeUser <USERNAME>
```

30.4. 配置智能路由器行为

在具有多个 KIE 服务器的集群环境中，默认行为是并行向每个 KIE 服务器发送请求，每个 KIE 服务器的主机将使用"round-robin"方法发送请求。在以下示例中，每个 KIE 服务器都使用相同的 KJAR，但每个 KJAR 版本都不同：

表 30.1. 示例环境

服务器名称	KJAR 版本	主机
kie-server1	kjar:1.0(alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=1.0)	129.0.1.1, 129.0.1.2, 129.0.1.3
kie-server2	kjar:2.0(alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=2.0)	129.0.2.1, 129.0.2.2, 129.0.2.3
kie-server3	kjar:3.0(alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=3.0)	129.0.3.1, 129.0.3.2, 129.0.3.3

如果您发送请求，请求将发送到 kie-server1(129.0.1.2), kie-server2(129.0.2.3), 和 kie-server3(129.0.3.1)。

如果您发送第二个请求，则该请求将发送到每个 KIE 服务器的下一主机。例如：kie-server1(129.0.1.3)、kie-server2(129.0.2.1) 和 kie-server3(129.0.3.2)。

智能路由器有三个组件，您可以修改它们来更改此行为：

ContainerResolver

负责查找与服务器交互时要使用的容器 ID 的组件。

RestrictionPolicy

负责禁止智能路由器使用特定端点的组件。

ConfigRepository

负责维护智能路由器配置的组件。这主要与路由表相关。

IdentityService

负责使用您自己的身份提供程序的组件。这适用于 KIE 服务器实例。

智能路由器使用 ServiceLoader 程序来实现以下组件：

ContainerResolver

`META-INF/services/org.kie.server.router.spi.ContainerResolver`

RestrictionPolicy

META-INF/services/org.kie.server.router.spi.RestrictionPolicy

ConfigRepository

META-INF/services/org.kie.server.router.spi.ConfigRepository

IdentityService

META-INF/services/org.kie.server.router.identity.IdentityService

例如，对于以上情况，您可以自定义 **ContainerResolver**，使智能路由器搜索整个可用 KIE 服务器中的最新版本 **KJAR** 进程，并始终以该进程启动。这种情况意味着每个 KIE 服务器都托管一个 **KJAR**，每个版本都会共享相同的别名。

由于 **Smart Router** 是一个可执行 **jar**，所以要包括扩展，您需要修改命令。例如：

```
java -cp LOCATION/router-ext-7.13.5.redhat-00002.jar:rhcam-7.13.5-smart-router.jar
org.kie.server.router.KieServerRouter
```

启动该服务后，您会看到日志输出，说明用于组件的实现：

```
Mar 01, 2017 1:47:10 PM org.kie.server.router.KieServerRouter <init>
INFO: KIE Server router repository implementation is InMemoryConfigRepository
Mar 01, 2017 1:47:10 PM org.kie.server.router.proxy.KieServerProxyClient <init>
INFO: Using 'LatestVersionContainerResolver' container resolver and restriction policy
'ByPassUserNotAllowedRestrictionPolicy'
Mar 01, 2017 1:47:10 PM org.xnio.Xnio <clinit>
INFO: XNIO version 3.3.6.Final
Mar 01, 2017 1:47:10 PM org.xnio.nio.NioXnio <clinit>
INFO: XNIO NIO Implementation Version 3.3.6.Final
Mar 01, 2017 1:47:11 PM org.kie.server.router.KieServerRouter start
INFO: KieServerRouter started on localhost:9000 at Wed Mar 01 13:47:11 CET 2017
```

30.5. 使用配置文件配置智能路由器设置

除了在命令行中配置智能路由器设置外，您还可以使用配置文件。在这种情况下，命令行终端和服务端日志中没有包括任何密码的设置。

流程

1. 创建配置文件。此文件可以包含 **属性=value** 格式的任意数量的行。

该文件可以包含以下任何属性：所有属性都是可选的。

表 30.2. 智能路由器配置文件中支持的属性

属性名称	描述	默认值
org.kie.server.router.id	智能路由器的标识符，用于识别其他组件。	N/A
org.kie.server.router.name	智能路由器的名称，用于识别其他组件。	N/A
org.kie.server.router.host	运行智能路由器的机器的主机名	localhost
org.kie.server.router.port	传入 HTTP 连接的端口。如果配置 TLS 连接，您可以将此属性设置为 0 以禁用 HTTP 连接。	9000
org.kie.server.router.url.external	用于访问智能路由器的外部 URL	N/A
org.kie.server.router.tls.port	传入 TLS 连接的端口	N/A
org.kie.server.router.tls.keystore	TLS 连接的密钥存储文件	N/A
org.kie.server.router.tls.keystore.password	用于 TLS 连接的密钥存储密码	N/A
org.kie.server.router.tls.keystore.keyalias	引用密钥存储中的 TLS 证书的别名名称	N/A
org.kie.server.router.repo	用于存储当前软件仓库的目录	当前工作目录
org.kie.router.identity.provider	可选的自定义供应商类，用于通过智能路由器验证 KIE 服务器实例。这个类必须实现 org.kie.server.router.identity.IdentityManager 接口，对于这个接口的源代码，请参阅 GitHub 存储库 。	N/A
org.kie.server.controller	连接到控制器的 URL	N/A
org.kie.server.controller.user	用于连接控制器的用户名	kieserver

属性名称	描述	默认值
org.kie.server.controller.password	连接到控制器的密码	kieserver1!
org.kie.server.controller.token	用于连接控制器的身份验证令牌	N/A
org.kie.server.controller.retry.interval	如果控制器失败，则重试到控制器的连接的时间间隔（以秒为单位）	10
org.kie.server.controller.retry.limit	连接控制器的最大重试次数（如果失败）	infinite
org.kie.server.router.config.watcher.enabled	如果设置为 true ，智能路由器会定期扫描配置文件并应用任何更改	false
org.kie.server.router.config.watcher.interval	重新扫描配置文件的时间间隔（以秒为单位）	5
org.kie.server.router.management.password	如果设置为 true ，智能路由器需要一个密码来验证来自 KIE 服务器的连接	false

2.

使用命令行启动智能路由器：

```
java -Dorg.kie.server.router.config.file=<CONFIG_FILE> -jar rhpam-7.13.5-smart-router.jar
```

将 **<CONFIG_FILE >** 替换为配置文件的名称。

第 31 章 配置 QUARTZ 计时器服务

当您在集群中运行 KIE 服务器时，您可以配置 Quartz 计时器服务。

在应用服务器上配置数据库之前，您必须准备 Quartz 的数据库，以创建 Quartz 表，该表将保存计时器数据和 Quartz 定义文件。

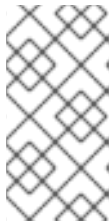
先决条件

- 一个受支持的非 JTA 数据源连接到您的应用服务器，如 PostgreSQL 数据源。

流程

1. 在您的数据库中创建 Quartz 表，以启用计时器事件以使用 DDL 脚本进行您的数据库同步。

DDL 脚本位于 `QUARTZ_HOME/docs/dbTables` 中提取的补充 ZIP 存档中。



注意

包含单词 `drop`（如 `quartz_tables_drop_db2.sql`）的脚本在创建前丢弃 Quartz 表。

2. 在 `JBOSS_HOME/MODE/configuration/` 目录下创建 `Quartz-definition.properties`，并添加以下示例内容：

```

#=====
# Configure Main Scheduler Properties
#=====
org.quartz.scheduler.instanceName = jBPMClusteredScheduler
org.quartz.scheduler.instanceId = AUTO
#=====
# Configure ThreadPool
#=====
org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 5
org.quartz.threadPool.threadPriority = 5
#=====

```

```

# Configure JobStore
#=====

org.quartz.jobStore.misfireThreshold = 60000
org.quartz.jobStore.class=org.quartz.impl.jdbcjobstore.JobStoreCMT
org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDeleg
egate
org.quartz.jobStore.useProperties=false
org.quartz.jobStore.dataSource=managedDS
org.quartz.jobStore.nonManagedTXDataSource=notManagedDS
org.quartz.jobStore.tablePrefix=QRTZ_
org.quartz.jobStore.isClustered=true
org.quartz.jobStore.clusterCheckinInterval = 20000
#=====

# Configure Datasources
#=====

org.quartz.dataSource.managedDS.jndiURL=jboss/datasources/psbpmsDS
org.quartz.dataSource.notManagedDS.jndiURL=jboss/datasources/quartzNotManaged
DS
# Note the configured data sources that accommodate the two Quartz schemes at the
very end of the file.

```



重要

集群发现的建议间隔为 20 秒，它在 `quartz-definition.properties` 文件的 `org.quartz.jobStore.clusterCheckinInterval` 属性中设置。考虑对您的系统的性能影响，并根据需要修改设置。

3. 提供到 `org.quartz.properties` 属性中的 `quartz-definition.properties` 文件的绝对路径。
4. 可选：要为 Quartz 触发器配置重试次数和延迟数，请更新以下系统属性：
 - `org.jbpm.timer.quartz.retries` (default value is 5)
 - `org.jbpm.timer.quartz.delay` (默认值为 1000)



注意

默认情况下，Qartz 需要两个数据源：

- 受管数据源，参与流程引擎的事务。
- 非受管数据源查找计时器，以便在不处理任何事务的情况下触发

Red Hat Process Automation Manager 商业应用程序假定 Quartz 数据库(schema) 将与 Red Hat Process Automation Manager 表在一起，因此生成用于 Quartz 事务操作的数据源。

必须配置其他（非事务性）数据源，但应指向与主数据源相同的数据库。

第 32 章 其他资源

- [在 Red Hat JBoss EAP 7.4 上安装和配置 Red Hat Process Automation Manager](#)
- [规划 Red Hat Process Automation Manager 安装](#)
- [使用 Operator 在 Red Hat OpenShift Container Platform 4 上部署 Red Hat Process Automation Manager 环境](#)
- [使用模板在 Red Hat OpenShift Container Platform 3 上部署 Red Hat Process Automation Manager 环境](#)

部分 IV. 在 RED HAT JBOSS WEB SERVER 上安装和配置 RED HAT PROCESS AUTOMATION MANAGER

本文档论述了如何在 JBoss Web 服务器上安装 Red Hat Process Automation Manager 7.13。



注意

在 Red Hat JBoss Web Server 和 Apache Tomcat 上对 Red Hat Process Automation Manager 的支持现已处于维护阶段。红帽将继续在这些平台上支持 Red Hat Process Automation Manager，但有以下限制：

- 红帽不会发布新的认证或软件功能。
- 红帽只会发布具有重大影响和关键任务程序错误修复补丁的合格安全修补程序。

未来，红帽可能会将客户定向到与红帽混合云策略兼容的新平台和产品组件。

先决条件

- 您已查看 [规划 Red Hat Process Automation Manager 安装中的信息](#)。
- 已安装 Red Hat JBoss Web Server 5.5.1。有关安装 Red Hat JBoss Web Server 的详情，请查看 [Red Hat JBoss Web Server 安装指南](#)。

第 33 章 关于 RED HAT PROCESS AUTOMATION MANAGER

Red Hat Process Automation Manager 是用于创建业务自动化应用程序和微服务的红帽中间件平台。它使企业业务和 IT 用户能够记录、模拟、管理、自动化和监控业务流程和策略。它旨在使商业和 IT 用户能够更有效地合作，从而轻松、快速地更改业务应用程序。

该产品由 **Business Central** 和 **KIE 服务器** 组成。

- **KIE 服务器**是执行规则和其他工件的服务器。它用于实例化和执行规则并解决规划问题。**KIE 服务器**为业务资产提供运行时环境，并访问存储在资产存储库中的数据（知识库）。

- **Business Central** 是用来创建和管理 **KIE 服务器**执行的业务规则的图形用户界面。它可让您执行以下任务：
 - 创建、管理和编辑您的规则、流程和相关资产。

 - 管理连接的 **KIE 服务器**实例及其 **KIE 容器**（部署单元）。

 - 针对连接到 **Business Central** 的 **KIE 服务器**实例中的流程和任务执行运行时操作。

Business Central 也作为独立 **JAR** 文件提供。您可以使用 **Business Central** 独立 **JAR** 文件来运行 **Business Central**，而无需将其部署到应用服务器。

Red Hat JBoss Web Server 是一个企业就绪的 **Web 服务器**，专为基于 **Tomcat** 的中型和大型应用程序设计。**红帽 JBoss Web 服务器**为组织提供了用于 **Java 服务器页面(JSP)**和 **Java Servlet** 技术、**PHP** 和 **CGI** 的单一部署平台。

在**红帽 JBoss Web 服务器**安装中，您可以安装 **KIE 服务器**和**无头进程自动化管理器**控制器。或者，您可以运行独立 **Business Central JAR** 文件。

本文档解释了如何在 **Red Hat JBoss Web Server** 实例中安装 **Red Hat Process Automation Manager**。

有关如何在其他环境中安装 Red Hat Process Automation Manager 的说明，请参阅以下文档：

- [在 Red Hat JBoss EAP 7.4 上安装和配置 Red Hat Process Automation Manager](#)
- [在 IBM WebSphere Application Server 上安装和配置 KIE 服务器](#)
- [在 Oracle WebLogic 服务器中安装和配置 KIE 服务器](#)
- [使用 Operator 在 Red Hat OpenShift Container Platform 4 上部署 Red Hat Process Automation Manager 环境](#)
- [使用模板在 Red Hat OpenShift Container Platform 3 上部署 Red Hat Process Automation Manager 环境](#)

有关支持的组件的详情，请查看以下文档：

- [Red Hat Process Automation Manager 和 Maven 库版本之间的映射是什么？](#)
- [Red Hat Process Automation Manager 7 支持的配置](#)

33.1. RED HAT PROCESS AUTOMATION MANAGER 组件

该产品由 Business Central 和 KIE 服务器组成。

- **Business Central** 是您创建和管理业务规则的图形用户界面。您可以在 Red Hat JBoss EAP 实例或 Red Hat OpenShift Container Platform(OpenShift)上安装 Business Central。

Business Central 也作为独立 JAR 文件提供。您可以使用 Business Central 独立 JAR 文件运行 Business Central，而无需将其部署到应用服务器。
- **KIE 服务器**是执行规则和其他工件的服务器。它用于实例化和执行规则并解决规划问题。您可以在红帽 JBoss EAP 实例中安装 KIE 服务器（在红帽 JBoss EAP 集群中），在 OpenShift

上、Oracle WebLogic 服务器实例、IBM WebSphere Application Server 实例或作为 Spring Boot 应用程序的一部分。

您可以将 KIE 服务器配置为在受管或非受管模式下运行。如果 KIE 服务器为非受管，您必须手动创建和维护 KIE 容器（部署单元）。KIE 容器是项目的特定版本。如果管理 KIE 服务器，则“流程自动化管理器”控制器管理 KIE 服务器配置，并与流程自动化管理器控制器交互以创建和维护 KIE 容器。

在红帽 JBoss Web 服务器安装中，您可以安装 KIE 服务器和无头进程自动化管理器控制器。或者，您可以运行独立 Business Central JAR 文件。

33.2. RED HAT PROCESS AUTOMATION MANAGER 角色和用户

要访问 Business Central 或 KIE 服务器，您必须在服务器启动前创建用户并为其分配适当的角色。安装 Business Central 或 KIE 服务器时，您可以创建用户和角色。

如果 Business Central 和 KIE 服务器都在单个实例上运行，则为 Business Central 进行身份验证的用户也可以访问 KIE 服务器。

然而，如果 Business Central 和 KIE 服务器在不同的实例上运行，则必须单独验证用于 Business Central 的用户才能访问 KIE 服务器。例如，如果在 Business Central 上进行身份验证但未在 KIE Server 进行身份验证的用户，如果在 Business Central 中查看或管理进程定义，则 401 错误会在日志文件中记录，而无效凭证来从远程服务器加载数据。请联系您的系统管理员。信息会出现在 Business Central 中。

本节论述了 Red Hat Process Automation Manager 用户角色。



注意

Business Central 保留了管理员、分析师、开发人员、经理、process-admin、user 和 rest-all 角色。kie-server 角色保留用于 KIE 服务器。因此，根据 Business Central、KIE 服务器还是安装这两者，可用角色可能会有所不同。

- **管理：**具有管理员角色的用户是 Business Central 管理员。他们可以管理用户、创建、克隆和管理存储库。他们可以完全访问，以便对应用程序进行必要的更改。具有 admin 角色的用户可以访问 Red Hat Process Automation Manager 中所有区域。
- **分析师：**具有 analyst 角色的用户可以访问所有高级功能。它们可以建模和执行其项目。但

是，这些用户无法在 **Design → Projects** 视图中的空格中添加投稿者。对于管理员的 **Deploy → Execution Servers** 视图的访问，它不适用于具有 **analyst** 角色的用户。但是，当他们访问 **Library** 视角时，这两个用户可使用 **Deploy** 按钮。

- **开发人员**：具有 **开发人员** 角色的用户可以访问几乎所有的功能，并可管理规则、模型、流程、表单和仪表板。它们可以管理资产存储库，他们可以创建、构建和部署项目。只有某些管理功能（如创建和克隆新存储库）才会从具有 **developer** 角色的用户中隐藏。
- **管理器**：具有 **manager** 角色的用户可以查看报告。这些用户通常对有关业务流程及其性能、业务指标和其他业务相关的报告的统计信息感兴趣。具有此角色的用户只能访问进程和任务报告。
- **process-admin**：具有 **process-admin** 角色的用户是业务流程管理员。他们可以完全访问业务流程、业务任务和执行错误。这些用户也可以查看业务报告，并有权访问 **Task Inbox** 列表。
- **用户**：具有 **用户** 角色的用户可以处理 **Task Inbox** 列表，其中包含目前正在运行进程的业务任务。具有此角色的用户可以查看进程和任务报告并管理进程。
- **REST-all**：具有 **restall** 角色的用户可以访问 **Business Central REST** 功能。
- **kie-server**：具有 **kie-server** 角色的用户可以访问 **KIE 服务器 REST** 功能。该角色是必需的，用户必须有权访问 **Business Central** 中的 **管理** 和 **跟踪** 视图。

第 34 章 下载 RED HAT PROCESS AUTOMATION MANAGER 安装文件

您可以使用安装程序 JAR 文件或可部署 ZIP 文件来安装 Red Hat Process Automation Manager。您可以在互动或命令行界面(CLI)模式下运行安装程序。或者，您可以提取并配置 Business Central 和 KIE 服务器可部署 ZIP 文件。如果您想在不将其部署到应用程序服务器的情况下运行 Business Central，请下载 Business Central Standalone JAR 文件。

下载符合您环境和安装要求的 Red Hat Process Automation Manager 发行版本。

流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：

- 产品：流程自动化管理器

- Version: 7.13.5

2. 根据您的首选安装方法下载以下产品分发之一：



注意

您只需要下载这些发行版之一。

- 如果要使用安装程序在 Red Hat JBoss Web Server 上安装 Red Hat Process Automation Manager，请下载 Red Hat Process Automation Manager 7.13.5 Installer (rhpam-installer-7.13.5.jar)。安装程序图形用户界面帮助您完成安装过程。
- 要使用可部署的 ZIP 文件在红帽 JBoss Web 服务器上安装 KIE 服务器，请下载以下文件：
 - Red Hat Process Automation Manager 7.13.5 Add Ons (rhpam-7.13.5-add-ons.zip)
 -

Red Hat Process Automation Manager 7.13.5 Maven Repository (rhpam-7.13.5-maven-repository.zip)

- 要运行 **Business Central**，无需将它部署到应用服务器，请下载 **Red Hat Process Automation Manager 7.13.5 Business Central Standalone (rhpam-7.13.5-business-central-standalone.jar)**。

第 35 章 使用 RED HAT PROCESS AUTOMATION MANAGER 安装程序

这部分论述了如何使用安装程序 JAR 文件安装 Business Central 和 KIE 服务器。JAR 文件是一个可执行文件，可在现有 Red Hat JBoss Web Server 5.5.1 服务器安装中安装 Red Hat Process Automation Manager。您可以在互动或命令行界面(CLI)模式下运行安装程序。

后续步骤：

按照以下部分中的说明操作：

- [第 35.1 节 “在互动模式中使用安装程序”](#)
- [第 35.2 节 “在 CLI 模式中使用安装程序”](#)

35.1. 在互动模式中使用安装程序

Red Hat Process Automation Manager 的安装程序是一个可执行 JAR 文件。您可以使用它在现有 Red Hat JBoss Web Server 5.5.1 服务器安装中安装 Red Hat Process Automation Manager。



注意

为安全起见，您应该以非 root 用户身份运行安装程序。

先决条件

- **Red Hat Process Automation Manager 7.13.5 安装程序 已下载。** 具体说明请查看 [第 34 章 下载 Red Hat Process Automation Manager 安装文件](#)。
- **已安装受支持的 JDK。** 有关支持的 JDK 列表，请参阅 [Red Hat Process Automation Manager 7 支持的配置](#)。
- **受支持的红帽 JBoss Web 服务器 5.5.1 服务器安装可用。**
- **需要足够的用户权限以完成安装。**



注意

确保您已经使用具有 Tomcat 写入权限的用户登录。



JAR 二进制文件包含在 \$PATH 环境变量中。在 Red Hat Enterprise Linux 上，它包含在 java-\$JAVA_VERSION-openjdk-devel 软件包中。



注意

Red Hat Process Automation Manager 旨在使用 UTF-8 编码。如果底层 JVM 使用了不同的编码系统，则可能发生意外错误。为确保 JVM 使用 UTF-8，请使用 "-Dfile.encoding=UTF-8" 系统属性。有关系统属性列表，请查看 [附录 B, Business Central 系统属性](#)。

流程

1.

在终端窗口中，导航到您下载安装程序 JAR 文件并输入以下命令的目录：

```
java -jar rhpam-installer-7.13.5.jar
```



注意

在 Windows 上运行安装程序时，可能会提示您在安装过程中提供管理员凭证。要防止这个要求，请在安装命令中添加 `izpack.mode=privileged` 选项：

```
java -Dizpack.mode=privileged -jar  
rhpam-installer-7.13.5.jar
```

另外，当在 32 位 Java 虚拟机上运行安装程序时，您可能会遇到内存限制。要防止这个问题，请运行以下命令：

```
java -XX:MaxHeapSize=4g -jar  
rhpam-installer-7.13.5.jar
```

图形安装程序显示启动画面和许可证协议页面。

2.

单击 **我接受此许可协议的条款**，然后单击下一步。

3. 指定您要安装 Red Hat Process Automation Manager 的 Red Hat JBoss Web Server 5.5.1 服务器主页，并单击下一步。

4. 选择您要安装的组件，然后点 Next。

您不能在红帽 JBoss Web 服务器上安装 Business Central。您只能在红帽 JBoss EAP 上安装它。但是，您可以在 Red Hat JBoss Web Server 上安装 KIE 服务器和无头进程自动化管理器控制器。无头进程自动化管理器控制器用于管理 KIE 服务器。如果您计划管理多个 KIE 服务器实例，请安装无头进程自动化管理器控制器。

5. 创建一个用户，然后单击下一步。默认情况下，如果您在同一容器中同时安装 Business Central 和 KIE 服务器，则会为新用户提供管理员、kie-server 和 rest-all 角色。如果您只安装 KIE 服务器，则向用户提供 kie-server 角色。需要 kie-server 角色来访问 KIE 服务器 REST 功能。



注意

确保指定的用户名与现有用户、角色或组不同。例如，不要创建用户名为 admin 的用户。

密码必须至少包含八个字符，且必须至少包含一个数字和一个非字母数字字符，但不包括 和 (ampersand)。

记录用户名和密码。您将需要他们访问 Business Central 和 KIE 服务器。

6. 在 Installation Overview 页面中，查看您要安装的组件，然后点 Next 开始安装。

7. 安装完成后，单击 下一步。

8. 如果安装了 KIE 服务器，则"配置运行时"步骤在"组件安装"下显示。在 Configure Runtime Environment 页面中，选择执行默认安装或执行高级配置。

如果您选择 执行高级配置，可以选择配置数据库设置或自定义某些 KIE 服务器选项。

9. 如果选择了 Customize database settings，在 JDBC Drive Configuration 页面上指定数

据源 JDBC 驱动程序厂商，请选择一个或多个驱动程序 JAR 文件，然后单击 **Next**。

数据源是一个对象，它允许 Java 数据库连接(JDBC)客户端（如应用服务器）与数据库建立连接。应用程序在 Java 命名和目录接口(JNDI)树或本地应用程序上下文中查找数据源，并请求数据库连接来检索数据。您必须为 KIE 服务器配置数据源，以确保在服务器和指定数据库之间进行正确的数据交换。

10.

如果您选择了 **Customize KIE 服务器属性**，在 **KIE Server Properties Configuration** 页面中更改以下任何属性：

- 更改 KIE 服务器 ID 的值，以更改 KIE 服务器属性的名称。
- 取消选择您想禁用的任何 KIE 服务器功能。

11.

单击 **Next** 来配置运行时环境。

12.

当处理在屏幕顶部出现时，单击 **Next** 完成安装。

13.

可选：单击 **Generate Installation Script and Properties File** 将安装数据保存在 XML 文件中，然后点 **Done**。

安装程序会生成两个文件。**auto.xml** 文件可自动安装，**auto.xml.variables** 文件存储用户密码和其他敏感变量。使用 **auto.xml** 文件在具有相同类型服务器的多个系统中重复 Red Hat Process Automation Manager 安装，以及与原始安装相同的配置。如果需要，更新 **auto.xml** 文件中的 **installpath** 参数。要使用 XML 文件执行安装，请输入以下命令：

```
java -jar rhpam-installer-7.13.5.jar <path-to-auto.xml-file>
```

您已使用安装程序成功安装了 Red Hat Process Automation Manager。如果您只安装了 **Business Central**，请重复这些步骤以便在单独的服务器中安装 KIE 服务器。



注意

如果您使用 **Microsoft SQL Server**，请确保为您的数据库配置了适用的事务隔离。如果没有，您可能会遇到死锁。推荐的配置通过输入以下语句开启 **ALLOW_SNAPSHOT_ISOLATION** 和 **READ_COMMITTED_SNAPSHOT**：

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

35.2. 在 CLI 模式中使用安装程序

您可以使用命令行界面(CLI)来运行 **Red Hat Process Automation Manager** 安装程序。



注意

为安全起见，您应该以非 **root** 用户身份运行安装程序。

先决条件

- **Red Hat Process Automation Manager 7.13.5 安装程序** 已下载。具体说明请查看 [第 34 章 下载 Red Hat Process Automation Manager 安装文件](#)。
- 已安装受支持的 **JDK**。有关支持的 **JDK** 列表，请参阅 [Red Hat Process Automation Manager 7 支持的配置](#)。
- 受支持的红帽 **JBoss Web 服务器 5.5.1** 服务器安装可用。
- 需要足够的用户权限以完成安装。



注意

确保您已经使用具有 **Tomcat** 写入权限的用户登录。

- **JAR** 二进制文件包含在 **\$PATH** 环境变量中。在 **Red Hat Enterprise Linux** 上，它包含在 **java-\$JAVA_VERSION-openjdk-devel** 软件包中。



注意

Red Hat Process Automation Manager 旨在使用 UTF-8 编码。如果底层 JVM 使用了不同的编码系统，则可能发生意外错误。为确保 JVM 使用 UTF-8，请使用 "-Dfile.encoding=UTF-8" 系统属性。有关系统属性列表，请查看 [附录 B, Business Central 系统属性](#)。

流程

1. 在终端窗口中，进入下载安装程序文件的目录，并输入以下命令：

```
java -jar rhpam-installer-7.13.5.jar -console
```

命令行交互式过程将启动并显示最终用户许可证协议。

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

2. 阅读许可证协议，输入 **1**，然后按 **Enter** 继续：

```
Specify the home directory of one of the following servers: Red Hat JBoss EAP 7 or Red Hat JBoss Web Server 5. For more information, see https://access.redhat.com/articles/3405381[Red Hat Process Automation Manager 7 Supported Configurations].
```

3. 输入现有 **Red Hat JBoss Web Server 5.5.1** 安装的父目录。

安装程序将验证安装在提供的位置的位置。输入 **1** 确认并继续。

4. 按照安装程序中的说明完成安装。



注意

在创建用户名和密码时，请确保指定的用户名和密码与角色或组的任何已知标题不冲突。例如，如果有一个名为 **admin** 的角色，您不应创建用户名为 **admin** 的用户。

密码必须至少包含八个字符，且必须至少包含一个数字和一个非字母数字字符（不包括字符和）。

记录用户名和密码。您将需要他们访问 **Business Central** 和 **KIE 服务器**。

5. 安装完成后，您会看到此消息：

```
Would you like to generate an automatic installation script and properties file?
```

6. 输入 **y** 来创建包含安装数据的 **XML** 文件，或 **n** 完成安装。如果您输入 **y**，会提示您输入 **XML** 文件的路径。

7. 输入路径或者按 **Enter** 键接受推荐的路径。

安装程序会生成两个文件。**auto.xml** 文件可自动安装，**auto.xml.variables** 文件存储用户密码和其他敏感变量。在与原始安装相同的配置相同的服务器上，使用多个系统上的 **auto.xml** 文件轻松重复 **Red Hat Process Automation Manager** 安装。如果需要，更新 **auto.xml** 文件中的 **installpath** 参数。要使用 **XML** 文件执行安装，请输入以下命令：

```
java -jar rhpam-installer-7.13.5.jar <path-to-auto.xml-file>
```

8. 如果您只安装了 **KIE 服务器**，请重复这些步骤以在单独的服务器上安装无头进程自动化管理器控制器。



注意

如果您使用 **Microsoft SQL Server**，请确保为您的数据库配置了适用的事务隔离。如果没有，您可能会遇到死锁。推荐的配置通过输入以下语句开启

ALLOW_SNAPSHOT_ISOLATION 和 **READ_COMMITTED_SNAPSHOT** :

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON  
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

第 36 章 KIE SERVER ZIP 文件安装和配置

您可以使用来自 Red Hat Process Automation Manager 7.13.5 Add Ons (rhpm-7.13.5-add-ons.zip)文件的 rhpm-7.13.5-kie-server-jws.zip 文件安装 KIE 服务器，然后在客户门户网站上配置 Java 数据库连接(JDBC) Web 服务器数据源。

36.1. 从 ZIP 文件安装 KIE 服务器

KIE 服务器为业务资产提供运行时环境，并访问存储在资产存储库中的数据（知识库）。您可以使用 ZIP 文件在现有 Red Hat JBoss Web Server 5.5.1 服务器实例上安装 KIE 服务器。



注意

要使用安装程序 JAR 文件来安装 KIE 服务器，请参阅 [第 35 章 使用 Red Hat Process Automation Manager 安装程序](#)。

- 以下文件已下载，如 [第 34 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述：
 - Red Hat Process Automation Manager 7.13.5 Add Ons (rhpm-7.13.5-add-ons.zip)
 - Red Hat Process Automation Manager 7.13.5 Maven Repository (rhpm-7.13.5-maven-repository.zip)
- 受支持的红帽 JBoss Web 服务器 5.5.1 服务器安装可用。红帽 JBoss Web 服务器安装的基本目录称为 `JWS_HOME`。
- 需要足够的用户权限以完成安装。

流程

1. 提取 rhpm-7.13.5-add-ons.zip 文件。

2. 在提取的 `rhpm-7.13.5-add-ons.zip` 文件中提取以下文件：

- `rhpm-7.13.5-kie-server-jws.zip`
- `rhpm-7.13.5-process-engine.zip`

在以下说明中，包含提取的 `rhpm-7.13.5-kie-server-jws.zip` 文件的目录名为 `JWS_TEMP_DIR`，包含提取的 `rhpm-7.13.5-process-engine.zip` 文件名为 `ENGINE_TEMP_DIR`。

3. 将 `JWS_TEMP_DIR/rhpm-7.13.5-kie-server-jws/kie-server.war` 目录复制到 `JWS_HOME/tomcat/webapps` 目录。



注意

确定您复制的 Red Hat Process Automation Manager 部署的名称不会与 Red Hat JBoss Web Server 实例中的现有部署冲突。

4. 从 `kie-server.war` 文件夹删除 `.war` 扩展。
5. 将 `kie-tomcat-integration-7.67.0.Final-redhat-00024.jar` 文件从 `ENGINE_TEMP_DIR` 目录移到 `JWS_HOME/tomcat/lib` 目录。
6. 将 `jboss-jacc-api-<VERSION>.jar`, `slf4j-api-<VERSION>.jar`, `slf4j-jdk14-<VERSION>.jar` 文件从 `ENGINE_TEMP_DIR/lib` 目录移到 `JWS_HOME/tomcat/lib` 目录中，其中 `<VERSION>` 是工件文件名（在 `lib` 目录中）。
7. 在最后一个 Valve 定义后，将下面这一行添加到 `JWS_HOME/tomcat/conf/server.xml` 文件中的 `<host>` 元素：

```
<Valve className="org.kie.integration.tomcat.JACCValve" />
```

8. 在文本编辑器中打开 `JWS_HOME/tomcat/conf/tomcat-users.xml` 文件。

9.

将用户和角色添加到 `JWS_HOME/tomcat/conf/tomcat-users.xml` 文件。在以下示例中，`<ROLE_NAME>` 是 Red Hat Process Automation Manager 支持的角色。`<USER_NAME>` 和 `<USER_PWD>` 是您选择的用户名和密码：

```
<role rolename="<ROLE_NAME>"/>
<user username="<USER_NAME>" password="<USER_PWD>" roles="<ROLE_NAME>"/>
```

如果用户有多个角色，如下例所示，使角色与逗号分开：

```
<role rolename="admin"/>
<role rolename="kie-server"/>
<user username="rhpamUser" password="user1234" roles="admin,kie-server"/>
```

10.

在 `JWS_HOME/tomcat/bin` 目录中完成以下步骤：

- 在 Linux 或 UNIX 中，使用以下内容创建 `setenv.sh` 文件：

```
CATALINA_OPTS="-Xmx1024m -Dorg.jboss.logging.provider=jdk"
```

- 在 Windows 中，将以下内容添加到 `setenv.bat` 文件中：

```
set CATALINA_OPTS=-Xmx1024m -Dorg.jboss.logging.provider=jdk
```

注意

如果您使用 Microsoft SQL Server，请确定您的数据库有适当的事务隔离。如果没有，您可能会遇到死锁。推荐的配置通过输入以下语句开启 `ALLOW_SNAPSHOT_ISOLATION` 和 `READ_COMMITTED_SNAPSHOT`：

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

36.2. 配置 JDBC WEB 服务器数据源

Java 数据库连接(JDBC)是一种 API 规范，用于将 Java 编写的程序连接到数据库。数据源是一个对象，它允许 Java 数据库连接(JDBC)客户端（如应用服务器）与数据库建立连接。应用程序在 Java 命名和目录接口(JNDI)树或本地应用程序上下文中查找数据源，并请求数据库连接来检索数据。您必须为 KIE 服务器配置数据源，以确保在服务器和指定数据库之间进行正确的数据交换。

通常，使用 Red Hat Process Automation Manager 的解决方案在单个事务中管理多个资源。用于异步作业、事件和计时器的 JMS，例如：在确保数据原子性和一致性结果时，Red Hat Process Automation Manager 需要数据源中的 XA 驱动程序。如果监听器内存在针对不同架构的事务代码，或者从 jBPM 引擎提供的 hook 派生出来，则还需要一个 XA 驱动程序。

不要使用非 XA 数据源，除非您正面没有参与单一事务的多个资源。



注意

对于生产环境，请指定实际的数据源。不要在生产环境中使用示例数据源。

先决条件

- Red Hat Process Automation Manager 安装在 Red Hat JBoss Web Server 上。
- Red Hat Process Automation Manager 7.13.5 Maven Repository (rhpam-7.13.5-maven-repository.zip)和 Red Hat Process Automation Manager 7.13.x Add-Ons (rhpam-7.13.5-add-ons.zip)文件已下载，如 [第 34 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述。
- 您需要配置以下支持的数据库和休眠方法之一：
 - DB2: org.hibernate.dialect.DB2Dialect
 - MSSQL: org.hibernate.SQLServer 2012Dialect
 - MySQL: org.hibernate.dialect.MySQL5InnoDBDialect
 - MariaDB : org.hibernate.dialect.MySQL5InnoDBDialect
 - Oracle : org.hibernate.dialect.Oracle10gDialect

- PostgreSQL : org.hibernate.dialect.PostgreSQL82Dialect
- PostgreSQL 加上 : org.hibernate.dialect.PostgresPlusPlusDialect
- Sybase: org.hibernate.dialect.SybaseASE157Dialect

流程

1. 完成以下步骤以准备数据库：
 - a. 在临时目录中提取 rhpam-7.13.5-add-ons.zip, 如 TEMP_DIR。
 - b. 提取 TEMP_DIR/rhpam-7.13.5-migration-tool.zip。
 - c. 将您的当前目录更改为 TEMP_DIR/rhpam-7.13.5-migration-tool/ddl-scripts 目录。此目录包含多种数据库类型的 DDL 脚本。
 - d. 将您的数据库类型的 DDL 脚本导入到您要使用的数据库中, 例如 :

```
psql jbpm < /ddl-scripts/postgresql/postgresql-jbpm-schema.sql
```



注意

如果您与 Spring Boot 结合使用 PostgreSQL 或 Oracle, 则必须导入对应的 Spring Boot DDL 脚本, 如 /ddl-scripts/oracle/oracle-springboot-jbpm-schema.sql 或 /ddl-scripts/postgresql/postgresql-springboot-jbpm-schema.sql。

2. 提取 rhpam-7.13.5-maven-repository.zip 离线 Maven 存储库文件。
3. 将以下库从提取的 Maven 存储库复制到 JWS_HOME/tomcat/lib 文件夹, 其中 VERSION 是该库的版本 :

```
org/jboss/spec/javax/transaction/jboss-transaction-api_1.2_spec/{VERSION}/jboss-transaction-api_1.2_spec-{VERSION}.jar
org/jboss/integration/narayana-tomcat/{VERSION}/narayana-tomcat-{VERSION}.jar
org/jboss/narayana/jta/narayana-jta/{VERSION}/narayana-jta-{VERSION}.jar
org/jboss/jboss-transaction-spi/{VERSION}/jboss-transaction-spi-{VERSION}.jar
```

4.

将数据库 JDBC 驱动程序复制到 JWS_HOME/tomcat/lib 文件夹。

5.

在 JWS_HOME/tomcat/conf/context.xml 文件中配置 pooling XA 数据源：



注意

以下示例中的一些属性可能不适用于您的数据库服务器。检查 JDBC 驱动程序的文档，以确定要设置哪些属性。

a.

在没有池功能的情况下配置 XA 数据源。此 XA 数据源用于创建与目标数据库的新连接。在以下示例中，IXA 数据源是 xads，变量在表 36.1 “XA 数据源变量”中定义：

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
<Resource
auth="Container"
databaseName="${datasource.dbName}"
description="XA Data Source"
factory="org.apache.tomcat.jdbc.naming.GenericNamingResourcesFactory"
loginTimeout="0"
name="xads"
uniqueName="xads"
portNumber="${datasource.port}"
serverName="${datasource.hostname}"
testOnBorrow="false"
type="${datasource.class}"
url="${datasource.url}"
URL="${datasource.url}"
user="${datasource.username}"
password="${datasource.password}"
driverType="4"
schema="${datasource.schema}"
/>
</Context>
```

表 36.1. XA 数据源变量

变量	描述
<code><datasource.dbName></code>	数据库的名称。
<code><datasource.port></code>	数据库的端口号。
<code><datasource.hostname></code>	数据库主机的名称。
<code><datasource.class></code>	JDBC 驱动程序的 XADataSource 类。
<code><datasource.url></code>	JDBC 数据库连接 URL。对于某些数据库，URL 属性是 url ，其他数据库（如 H2 数据库）是 URL 。
<code><datasource.username></code>	数据库连接的用户名。
<code><datasource.password></code>	数据库连接的密码。
<code><datasource.schema></code>	数据库架构。

b.

配置一个池数据源，它依赖于 XA 数据源创建新连接。在本例中，数据源是 `poolingXaDs`，`<datasource.username >` 是数据库连接的用户名，`<datasource.password >` 是数据库连接的密码：

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
<Resource
name="poolingXaDs"
uniqueName="poolingXaDs"
auth="Container"
description="Pooling XA Data Source"
factory="org.jboss.narayana.tomcat.jta.TransactionalDataSourceFactory"
testOnBorrow="true"
transactionManager="TransactionManager"
transactionSynchronizationRegistry="TransactionSynchronizationRegistry"
type="javax.sql.XADataSource"
username="${datasource.username}"
password="${datasource.password}"
xaDataSource="xads"
/>
</Context>
```

现在，数据源位于 `java:comp/env/pool/poolingXaDs` JNDI 名称下，并通过 `org.kie.server.persistence.ds` 系统属性传递给 KIE 服务器，如后续步骤中所述。

**注意**

池数据源配置依赖于之前在 `kie-server` 应用中的 `context.xml` 文件中配置的额外资源，特别是 `TransactionManager` 和 `TransactionSynchronizationRegistry`。

6.

配置 KIE 服务器以使用数据源：

a.

在文本编辑器中打开以下脚本之一：

**注意**

`setenv.sh` 或 `setenv.bat` 脚本应已存在。但是，如果没有，请创建它。

•

对于 **Linux** 或 **Unix**：

```
JWS_HOME/tomcat/bin/setenv.sh
```

•

对于 **Windows**：

```
JWS_HOME/tomcat/bin/setenv.bat
```

b.

在 `CATALINA_OPTS` 中添加以下属性，其中 `<hibernate.dialect>` 是您数据库的 **Hibernate** 断处：

```
CATALINA_OPTS="-Xmx1024m
-Dorg.jboss.logging.provider=jdk
-Dorg.kie.server.persistence.ds=java:comp/env/poolingXaDs
-Dorg.kie.server.persistence.tm=JBossTS
-Dorg.kie.server.persistence.dialect=${<hibernate.dialect>}"
```

第 37 章 使用密钥存储保护密码

您可以使用密钥存储来加密用于 **Business Central** 和 **KIE 服务器** 之间的通信的密码。您应该同时对控制器和 **KIE 服务器** 密码进行加密。如果 **Business Central** 和 **KIE 服务器** 部署到不同的应用程序服务器，则两个应用服务器都应该使用密钥存储。

为您的密钥存储使用 **Java Cryptography Extension KeyStore(JCEKS)**，因为它支持对称密钥。使用 **KeyTool** 作为 **JDK** 安装的一部分，可以创建一个新的 **JCEKS**。



注意

如果没有使用 **JCEKS** 配置 **KIE 服务器**，则 **KIE 服务器** 密码以纯文本形式保存在系统属性中。

先决条件

- **KIE 服务器** 安装在 **Red Hat JBoss Web Server** 中。
- 已安装 **Java 8** 或更高版本。

流程

1. 在文本编辑器中打开 `JWS_HOME/tomcat/conf/tomcat-users.xml` 文件。
2. 使用 `kie-server` 角色添加 **KIE 服务器** 用户到 `JWS_HOME/tomcat/conf/tomcat-users.xml` 文件。在以下示例中，将 `<USER_NAME>` 和 `<PASSWORD>` 替换为您选择的用户名和密码。

```
<role rolename="kie-server"/>
<user username="<USER_NAME>" password="<PASSWORD>" roles="kie-server"/>
```

3. 要使用 **KeyTool** 创建 **JCEKS**，请在 **Java 8** 主目录中输入以下命令：

```
$<JAVA_HOME>/bin/keytool -importpassword -keystore <KEYSTORE_PATH> -
keypass <ALIAS_KEY_PASSWORD> -alias <PASSWORD_ALIAS> -storepass
<KEYSTORE_PASSWORD> -storetype JCEKS
```

在本例中，替换以下变量：

- **<KEYSTORE_PATH > : 存储密钥存储的路径**
 - **<KEYSTORE_PASSWORD> : 密钥存储密码**
 - **<ALIAS_KEY_PASSWORD > : 用于访问使用别名存储的值的密码**
 - **<PASSWORD_ALIAS > : 条目的别名**
4. 在提示时，输入您创建的 KIE 服务器用户的密码。
5. 要设置系统属性，请在 *JWS_HOME*/tomcat/bin 目录中完成这些步骤之一，并替换下表所述的变量：



注意

如果在来自红帽 JBoss Web 服务器的独立实例中安装了 Business Central 或独立控制器，请不要将 `kie.keystore.key.server.alias` 和 `kie.keystore.key.pwd` 属性添加到 `CATALINA_OPTS`。

- 在 Linux 或 UNIX 中，使用以下内容创建 `setenv.sh` 文件：

```
set CATALINA_OPTS="
-Dkie.keystore.keyStoreURL=<KEYSTORE_URL>
-Dkie.keystore.keyStorePwd=<KEYSTORE_PWD>
-Dkie.keystore.key.server.alias=<KEY_SERVER_ALIAS>
-Dkie.keystore.key.server.pwd=<KEY_SERVER_PWD>
-Dkie.keystore.key.ctrl.alias=<KEY_CONTROL_ALIAS>
-Dkie.keystore.key.ctrl.pwd=<KEY_CONTROL_PWD>
```

- 在 Windows 中，将以下内容添加到 `setenv.bat` 文件中：

```
set CATALINA_OPTS="
-Dkie.keystore.keyStoreURL=<KEYSTORE_URL>
```

```

-Dkie.keystore.keyStorePwd=<KEYSTORE_PWD>
-Dkie.keystore.key.server.alias=<KEY_SERVER_ALIAS>
-Dkie.keystore.key.server.pwd=<KEY_SERVER_PWD>
-Dkie.keystore.key.ctrl.alias=<KEY_CONTROL_ALIAS>
-Dkie.keystore.key.ctrl.pwd=<KEY_CONTROL_PWD>

```

表 37.1. 用于加载 KIE 服务器 JCEKS 的系统属性

系统属性	占位符	描述
kie.keystore.keyStoreURL	<KEYSTORE_URL>	要使用的 JCEKS 的 URL，如 file:///home/kie/keystores/keystore.jceks
kie.keystore.keyStorePwd	<KEYSTORE_PWD>	JCEKS 的密码
kie.keystore.key.server.alias	<KEY_SERVER_ALIASS>	保存密码的 REST 服务的密钥别名
kie.keystore.key.server.pwd	<KEY_SERVER_PWD>	使用存储的密码的 REST 服务别名的密码
kie.keystore.key.ctrl.alias	<KEY_CONTROL_ALIASS>	存储密码的默认 REST Process Automation Controller 的密钥别名
kie.keystore.key.ctrl.pwd	<KEY_CONTROL_PWD>	默认 REST Process Automation Controller 的别名密码，其存储的密码

6. 启动 KIE 服务器以验证配置。

第 38 章 验证 KIE 服务器安装

验证正确安装了 KIE 服务器。

先决条件

- 已安装并配置了 KIE 服务器。

流程

1. 要启动 KIE 服务器，请在 `JWS_HOME/tomcat/bin` 目录中输入以下命令：
 - 在 Linux 或基于 UNIX 的系统中：

```
┆ $ ./startup.sh
```
 - 在 Windows 中：

```
┆ startup.bat
```
2. 几分钟后，检查 `JWS_HOME/tomcat/logs` 目录中的文件并更正任何错误。
3. 要验证 KIE 服务器是否正常工作，在网页浏览器中输入 `http://localhost:8080/kie-server/services/rest/server`。
4. 输入存储在 `tomcat-users.xml` 文件中的用户名和密码。

第 39 章 下载并安装无头进程自动化管理器控制器

您可以将 KIE 服务器配置为在受管或非受管模式下运行。如果 KIE 服务器为非受管，您必须手动创建和维护 KIE 容器（部署单元）。如果管理 KIE 服务器，则“流程自动化管理器”控制器管理 KIE 服务器配置，并与流程自动化管理器控制器交互以创建和维护 KIE 容器。

流程自动化管理器控制器与 Business Central 集成。如果您安装 Business Central，请使用 [执行服务器](#) 页面来创建和维护 KIE 容器。但是，如果您不安装 Business Central，则可安装无头流程自动化管理器控制器，并使用 REST API 或 KIE 服务器 Java 客户端 API 与它交互。

先决条件

- 已下载 Red Hat Process Automation Manager 7.13.5 Add Ons (rhpm-7.13.5-add-ons.zip) 文件，如 [第 34 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述。
- Red Hat JBoss Web Server 5.5.1 服务器安装可用。红帽 JBoss Web 服务器安装的基本目录称为 `JWS_HOME`。
- 需要足够的用户权限以完成安装。

流程

1. 提取 rhpm-7.13.5-add-ons.zip 文件。rhpm-7.13.5-controller-jws.zip 文件位于提取的目录中。
2. 将 rhpm-7.13.5-controller-jws.zip 存档提取到临时目录中。在以下示例中，此目录名为 `TEMP_DIR`。
3. 将 `TEMP_DIR/rhpm-7.13.5-controller-jws.zip/controller.war` 目录复制到 `JWS_HOME/tomcat/webapps` 目录。



注意

确定您复制的 Red Hat Process Automation Manager 部署的名称不会与 Red Hat JBoss Web Server 实例中的现有部署冲突。

4. 从 controller .war 文件夹删除 .war 扩展。
5. 将 `TEMP_DIR/rhpam-7.13.5-controller-jws/SecurityPolicy/` 目录的内容复制到 `JWS_HOME/bin`
6. 当系统提示覆盖文件时，选择是。
7. 将 `kie-server` 角色和用户添加到 `JWS_HOME/tomcat/conf/tomcat-users.xml` 文件中。在以下示例中，`<USER_NAME>` 和 `<PASSWORD>` 是您选择的用户名和密码：

```
<role rolename="kie-server"/>
<user username="<USER_NAME>" password="<PASSWORD>" roles="kie-server"/>
```

8. 在运行 KIE 服务器的实例的 `JWS_HOME/tomcat/bin` 目录中完成以下任务之一：

- 在 Linux 或 UNIX 中，使用以下内容创建 `setenv.sh` 文件：

```
CATALINA_OPTS="-Xmx1024m
-Dorg.jboss.logging.provider=jdk
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.id=<KIE_SERVER_ID>
-Dorg.kie.server.location=http://<HOST>:<PORT>/kie-server/services/rest/server
-Dorg.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller"
```

- 在 Windows 中，将以下内容添加到 `setenv.bat` 文件中：

```
set CATALINA_OPTS=-Xmx1024m -Dorg.jboss.logging.provider=jdk
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.id=<KIE_SERVER_ID>
-Dorg.kie.server.location=http://<HOST>:<PORT>/kie-server/services/rest/server
-Dorg.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller
```

在前面的示例中，替换以下变量：

- 将 `<CONTROLLER_USER>` 和 `<CONTROLLER_PWD>` 替换为您在此流程前面定义的 `kie-server` 角色的用户名和密码。

- 将 `<KIE_SERVER_ID>` 替换为唯一标识符。
 - 将 `<CONTROLLER_HOST>:<CONTROLLER_PORT>` 替换为控制器的 IP 地址（主机和端口）。如果您为 KIE 服务器和控制器使用相同的服务器，`<CONTROLLER_HOST>:<CONTROLLER_PORT>` 是 `localhost:8080`。
9. 在运行无头进程自动化管理器控制器的实例的 `JWS_HOME/tomcat/bin` 目录中，使用以下内容创建一个可读的 `setenv.sh` 文件，其中 `<USERNAME>` 是 KIE 服务器用户，`<USER_PWD>` 是该用户的密码：
- ```
CATALINA_OPTS="-Dorg.kie.server.user=<USERNAME> -Dorg.kie.server.pwd=<USER_PWD>"
```
10. 要启动无头进程自动化管理器控制器，在 `JWS_HOME/tomcat/bin` 目录中输入以下命令：
- 在 Linux 或基于 UNIX 的系统中：

```
┌ $./startup.sh
```
  - 在 Windows 中：

```
┌ startup.bat
```
11. 几分钟后，检查 `JWS_HOME/tomcat/logs` 目录并更正任何错误。
12. 要验证无头进程自动化管理器控制器是否正常工作，在网页浏览器中输入 `http://<CONTROLLER_HOST>:<CONTROLLER_PORT>/controller/rest/controller/management/servers`。如果您为 KIE 服务器和控制器使用相同的服务器，`<CONTROLLER_HOST>:<CONTROLLER_PORT>` 是 `localhost:8080`。
13. 输入存储在 `tomcat-users.xml` 文件中的用户名和密码。



## 第 40 章 配置并运行独立 BUSINESS CENTRAL

您可以使用 Business Central 独立 JAR 文件运行 Business Central，而无需将其部署到应用服务器。您可以使用示例配置文件来启动 Business Central 独立 JAR 文件，也可以根据您的要求自定义 `sampfiles`。

**注意**

只有在 Red Hat Enterprise Linux 上运行时，才支持这个 JAR 文件。

**先决条件**

- Red Hat Process Automation Manager 7.13.5 Business Central Standalone (`rhpm-7.13.5-business-central-standalone.jar`)和 Red Hat Process Automation Manager 7.13.5 Add Ons (`rhpm-7.13.5-add-ons.zip`)文件已从 Red Hat Process Automation Manager 7.13 的 [Software Downloads](#) 页面下载，如 [第 34 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述。

**流程**

1. 将下载的 `rhpm-7.13.5-addons.zip` 提取到临时目录中。此存档包括 `rhpm-7.13.5-standalone-sample-configuration.zip` 文件。
2. 将 `rhpm-7.13.5-standalone-sample-configuration.zip` 文件提取到包含 `rhpm-7.13.5-business-central-standalone.jar` 文件的目录。`rhpm-7.13.5-standalone-sample-configuration.zip` 文件包含以下示例配置文件：

- `application-script.cli`: 添加用户和 kie 服务器系统属性的 Sample 脚本
- `kie-fs-realm-users`: Sample user data

您可以使用配置文件中提供的示例数据运行 `rhpm-7.13.5-business-central-standalone.jar` 文件，或者您可以根据您的要求自定义数据。

3. 要自定义配置数据，请完成以下步骤：
  - a. 编辑 `application-script.cli` 文件，使其包含具有 `admin`、`用户`、`rest-all`、`rest-client`

和 `kie-server` 角色的管理用户。在以下示例中，将 `<USERNAME>` 和 `<PASSWORD>` 替换为您要创建的用户的用户名和密码。

```
/subsystem=elytron/filesystem-realm=KieRealm:add-identity(identity=<USERNAME>)
/subsystem=elytron/filesystem-realm=KieRealm:set-password(identity=<USERNAME>,
clear={password="<PASSWORD>"})
/subsystem=elytron/filesystem-realm=KieRealm:add-identity-attribute(identity=
<USERNAME>, name=role, value=["admin","user","rest-all","rest-client","kie-server"])
```

4.

要运行 **Business Central** 独立 JAR 文件，请输入以下命令：

```
java -jar rhpam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli
```

5.

要在运行 JAR 文件时设置应用程序属性，请在命令中包含 `-D<PROPERTY>=<VALUE>`；参数，其中 `<PROPERTY>` 是受支持的应用程序属性的名称，`<VALUE>` 是属性值：

```
java -jar rhpam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli -
D<PROPERTY>=<VALUE> -D<PROPERTY>=<VALUE>
```

例如，要运行 **Business Central** 并以用户 `controllerUser` 身份连接到 KIE 服务器，请输入：

```
java -jar rhpam-7.13.5-business-central-standalone.jar \
--cli-script=application-script.cli \
-Dorg.kie.server.user=controllerUser \
-Dorg.kie.server.pwd=controllerUser1234
```

这样做可让您将容器部署到 KIE 服务器。如需更多信息，请参阅 [附录 A, Business Central 系统属性](#)。



#### 注意

要在 **Business Central** 中启用用户和组管理，将 `org.uberfire.ext.security.management.wildfly.cli.folderPath` 属性的值设置为 `kie-fs-realm-users`。

## 第 41 章 RED HAT PROCESS AUTOMATION MANAGER 的 MAVEN 设置和软件仓库

创建 Red Hat Process Automation Manager 项目时，B Business Central 使用为 Business Central 配置的 Maven 存储库。您可以通过修改 Maven 项目对象模型(POM)文件(POM)文件(pom.xml)文件，使用 Maven 全局或用户设置来指示所有 Red Hat Process Automation Manager 项目，从公共 Red Hat Process Automation Manager 存储库检索依赖项。您还可以将 Business Central 和 KIE 服务器配置为使用外部 Maven 存储库，或准备 Maven mirror 以供离线使用。

有关 Red Hat Process Automation Manager 打包和部署选项的更多信息，请参阅 [打包和部署 Red Hat Process Automation Manager 项目](#)。

### 41.1. 使用项目配置文件配置 MAVEN(POM.XML)

要使用 Maven 来构建和管理 Red Hat Process Automation Manager 项目，您必须创建和配置 POM 文件(pom.xml)。此文件包含项目的配置信息。如需更多信息，请参阅 [Apache Maven Project](#)。

#### 流程

1. 生成 Maven 项目。创建 Maven 项目时会自动生成 pom.xml 文件。
2. 编辑 pom.xml 文件，以添加更多依赖项和新存储库。

当您编译和打包您的项目时，Maven 从 Maven 存储库下载所有 JAR 文件和依赖 JAR 文件。

找到位于 [http://maven.apache.org/maven-v4\\_0\\_0.xsd](http://maven.apache.org/maven-v4_0_0.xsd) 的 pom.xml 文件的 schema。有关 POM 文件的更多信息，请参阅 [Apache Maven Project POM](#)。

### 41.2. 修改 MAVEN 设置文件

Red Hat Process Automation Manager 使用 Maven settings.xml 文件来配置它的 Maven 执行。您必须在 settings.xml 文件中创建并激活配置集，并声明 Red Hat Process Automation Manager 项目使用的 Maven 存储库。

有关 Maven settings.xml 文件的详情，请查看 [Apache Maven Project Setting Reference](#)。

#### 流程

1. 在 `settings.xml` 文件中，声明您的 Red Hat Process Automation Manager 项目使用的软件仓库。通常，这是您从红帽客户门户下载的 Red Hat Process Automation Manager Maven 存储库或 Red Hat Process Automation Manager Maven 存储库，以及用于您要使用的自定义工件的任何软件仓库。
2. 确保 Business Central 或 KIE 服务器已配置为使用 `settings.xml` 文件。例如，指定 `kie.maven.settings.custom=<SETTINGS_FILE_PATH>` 属性，其中 `<SETTINGS_FILE_PATH>` 是 `settings.xml` 文件的路径。

在红帽 JBoss Web Server 上，KIE Server 的 KIE Server 将 `-Dkie.maven.settings.custom=<SETTINGS_FILE_PATH>` 添加到 `setenv.sh` (Linux) 或 `setenv.bat` (Windows) 文件的 `CATALINA_OPTS` 部分。对于独立 Business Central，请输入以下命令：

```
java -jar rhpam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli -Dkie.maven.settings.custom=<SETTINGS_FILE_PATH>
```

### 41.3. 为 RED HAT PROCESS AUTOMATION MANAGER 添加 MAVEN 依赖项

要在 Red Hat Process Automation Manager 项目中使用正确的 Maven 依赖项，请将 Red Hat Business Automation 材料(BOM)文件添加到项目的 `pom.xml` 文件中。Red Hat Business Automation BOM 适用于 Red Hat Decision Manager 和 Red Hat Process Automation Manager。添加 BOM 文件时，项目中包含来自提供的 Maven 存储库的正确依赖项版本。

有关 Red Hat Automation BOM 的更多信息，请参阅 [Red Hat Process Automation Manager 和 Maven 库版本之间的映射是什么？](#)

#### 流程

1. 在 `pom.xml` 文件中声明 Red Hat Business Automation BOM：

```
<dependencyManagement>
<dependencies>
<dependency>
<groupId>com.redhat.ba</groupId>
<artifactId>ba-platform-bom</artifactId>
<version>7.13.5.redhat-00002</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
```

```
<dependencies>
<!-- Your dependencies -->
</dependencies>
```

2.

在 **<dependencies>** 标签中声明项目所需的依赖项。将产品 BOM 导入到项目中后，定义了面向用户的依赖项的版本，因此您无需指定这些 **<dependency>** 元素的 **<version>** 子元素。但是，您必须使用 **<dependency>** 元素来声明您要在项目中使用的依赖项。

3.

对于未在 **Business Central** 中编写的独立项目，请指定项目所需的所有依赖项。在您在 **Business Central** 中编写的项目中，**Be Business Central** 会自动提供基本决策引擎和流程引擎依赖项。

•

对于基本的 **Red Hat Process Automation Manager** 项目，根据您要使用的功能，声明以下依赖项：

#### 嵌入式进程引擎依赖项

```
<!-- Public KIE API -->
<dependency>
 <groupId>org.kie</groupId>
 <artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-flow</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-flow-builder</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-bpmn2</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-runtime-manager</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-persistence-jpa</artifactId>
```

```

</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-query-jpa</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-audit</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-kie-services</artifactId>
</dependency>

<!-- Dependency needed for default WorkItemHandler implementations. -->
<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-workitems-core</artifactId>
</dependency>

<!-- Logging dependency. You can use any logging framework compatible with
slf4j. -->
<dependency>
 <groupId>ch.qos.logback</groupId>
 <artifactId>logback-classic</artifactId>
 <version>${logback.version}</version>
</dependency>

```

- 对于使用 CDI 的 Red Hat Process Automation Manager 项目，您通常会声明以下依赖项：

#### 启用 CDI 的进程引擎依赖项

```

<dependency>
 <groupId>org.kie</groupId>
 <artifactId>kie-api</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>

```

```
<groupId>org.jbpm</groupId>
<artifactId>jbpm-services-cdi</artifactId>
</dependency>
```

- 对于基本的 Red Hat Process Automation Manager 项目，声明以下依赖项：

#### 嵌入式决策引擎依赖项

```
<dependency>
 <groupId>org.drools</groupId>
 <artifactId>drools-compiler</artifactId>
</dependency>

<!-- Dependency for persistence support. -->
<dependency>
 <groupId>org.drools</groupId>
 <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>
 <groupId>org.drools</groupId>
 <artifactId>drools-decisiontables</artifactId>
</dependency>
<dependency>
 <groupId>org.drools</groupId>
 <artifactId>drools-templates</artifactId>
</dependency>
<dependency>
 <groupId>org.drools</groupId>
 <artifactId>drools-scorecards</artifactId>
</dependency>

<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
 <groupId>org.kie</groupId>
 <artifactId>kie-ci</artifactId>
</dependency>
```

- 要使用 KIE 服务器，声明以下依赖项：

## 客户端应用程序 KIE 服务器依赖项

```
<dependency>
 <groupId>org.kie.server</groupId>
 <artifactId>kie-server-client</artifactId>
</dependency>
```

- 要为 Red Hat Process Automation Manager 创建远程客户端，请声明以下依赖项：

## 客户端依赖项

```
<dependency>
 <groupId>org.uberfire</groupId>
 <artifactId>uberfire-rest-client</artifactId>
</dependency>
```

- 在创建包含资产的 JAR 文件（如规则和进程定义）时，请将 Maven 项目的打包类型指定为 `kjar`，并使用 `org.kie:kie-maven-plugin` 来处理位于 `<project>` 元素下的 `kjar` 打包类型。在以下示例中，`${kie.version}` 是列出的 Maven 库版本，在 [Red Hat Process Automation Manager 和 Maven 库版本之间有什么映射？](#)

```
<packaging>kjar</packaging>
<build>
 <plugins>
 <plugin>
 <groupId>org.kie</groupId>
 <artifactId>kie-maven-plugin</artifactId>
 <version>${kie.version}</version>
 <extensions>true</extensions>
 </plugin>
 </plugins>
</build>
```

### 41.4. 准备 MAVEN 镜像存储库以离线使用

如果您的 Red Hat Process Automation Manager 部署无法对公共互联网进行传出访问，则必须使用



所有必要工件的镜像准备 Maven 存储库，并使此软件仓库可供您的环境使用。



### 注意

如果您的 Red Hat Process Automation Manager 部署连接到互联网，则不需要完成这个步骤。

### 先决条件

- 有一个可以传出访问公共互联网的计算机。

### 流程

1. 在与公共互联网传出连接的计算机上，完成以下步骤：
2. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：

- 产品：流程自动化管理器

- Version: 7.13.5

- a. 下载并解压缩 Red Hat Process Automation Manager 7.13.5 Offliner Content List (rhpm-7.13.5-offliner.zip) 产品 deliverable 文件。

- b. 将 rhpm-7.13.5-offliner.zip 文件的内容提取到任何目录中。

- c. 进入该目录并输入以下命令：

```
./offline-repo-builder.sh offliner.txt
```

此命令创建 存储库 子目录，并将必要的工件下载到此子目录中。这是镜像存储库。

如果消息报告一些下载失败，请再次运行相同的命令。如果再次下载失败，请联系红帽支持。

3.

如果您在 **Business Central** 外部开发服务且它们有额外的依赖项，请将依赖项添加到镜像存储库中。如果您将服务开发成 **Maven** 项目，您可以使用以下步骤自动准备这些依赖项。在计算机上完成与公共互联网传出连接的计算机上的步骤。

a.

创建本地 **Maven** 缓存目录(`~/.m2/repository`)的备份，然后清除该目录。

b.

使用 `mvn clean install` 命令构建项目的源。

c.

对于每个项目，输入以下命令来确保 **Maven** 下载项目的所有运行时依赖项：

```
mvn -e -DskipTests dependency:go-offline -f /path/to/project/pom.xml --batch-mode -Djava.net.preferIPv4Stack=true
```

将 `/path/to/project/pom.xml` 替换为项目 `pom.xml` 文件的路径。

d.

将本地 **Maven** 缓存目录的内容(`~/.m2/ repository`)复制到创建的存储库子目录。

4.

将存储库子目录的内容复制到您在其上部署 **Red Hat Process Automation Manager** 的计算机的目录中。该目录变为离线 **Maven mirror** 存储库。

5.

为您的 **Red Hat Process Automation Manager** 部署创建并配置 `settings.xml` 文件，如第 41.2 节“[修改 Maven 设置文件](#)”所述。

6.

在 `settings.xml` 文件中进行以下更改：

- 

在 `<profile>` 标签下，如果缺少 `<repositories>` 或 `<pluginRepositories>` 标签，请添加缺失的标签。

- 

在 `<repositories>` 下添加以下内容：

```
<repository>
```

```
<id>offline-repository</id>
<url>file:///path/to/repo</url>
<releases>
 <enabled>true</enabled>
</releases>
<snapshots>
 <enabled>false</enabled>
</snapshots>
</repository>
```

使用到本地 Maven mirror repository 目录的完整路径替换 /path/to/repo。

- 在 <pluginRepositories > 下添加以下内容：

```
<repository>
 <id>offline-plugin-repository</id>
 <url>file:///path/to/repo</url>
 <releases>
 <enabled>true</enabled>
 </releases>
 <snapshots>
 <enabled>false</enabled>
 </snapshots>
</repository>
```

使用到本地 Maven mirror repository 目录的完整路径替换 /path/to/repo。

## 第 42 章 从 GIT 存储库导入项目

**Git** 是一种分布式版本控制系统。它以提交对象的形式实施修订版本。当您更改保存到存储库时，会在 **Git** 存储库中创建新的提交对象。

**Business Central** 使用 **Git** 存储项目数据，包括规则和流程等资产。当您在 **Business Central** 中创建项目时，它会被添加到嵌入式 **Business Central** 中的 **Git** 存储库中。如果您在其他 **Git** 存储库中拥有项目，您可以通过 **Business Central** 空间将这些项目导入到 **Business Central Git** 存储库中。

### 先决条件

- **Red Hat Process Automation Manager** 项目存在于外部 **Git** 存储库中。
- 有对该外部 **Git** 存储库的读取访问权限所需的凭证。

### 流程

1. 在 **Business Central** 中，点击 **Menu** → **Design** → **Projects**。
2. 选择或创建您要导入项目的空间。默认空间为 **MySpace**。
3. 要导入项目，请执行以下操作之一：
  - 单击 **Import Project**。
  - 从下拉菜单中选择 **Import Project**。
4. 在 **Import Project** 窗口中，输入包含您要导入的项目的 **Git** 存储库的 **URL** 和凭证，然后单击 **Import**。这些项目添加到 **Business Central Git** 存储库，可在当前空间中可用。

## 第 43 章 集成 LDAP 和 SSL

通过 Red Hat Process Automation Manager, 您可以通过 Red Hat Single Sign-On 集成 LDAP 和 SSL。如需更多信息, 请参阅 [Red Hat Single Sign-On Server 管理指南](#)。

## 附录 B. BUSINESS CENTRAL 系统属性

本节中列出的 Business Central 系统属性传递到 standalone\*.xml 文件。

### Git 目录

使用以下属性设置 Business Central Git 目录的位置和名称：

- **org.uberfire.nio.git.dir:** Business Central Git 目录的位置。
- **org.uberfire.nio.git.dirname:** Business Central Git 目录的名称。默认值：`.niogit`。
- **org.uberfire.nio.git.ketch:** Enables 或 disable Git ketch。
- **org.uberfire.nio.git.hooks:** Git hook 目录的位置。

### git over HTTP

使用以下属性配置通过 HTTP 对 Git 存储库的访问：

- **org.uberfire.nio.git.proxy.ssh.over.http:** 指定 SSH 是否应使用 HTTP 代理。默认值：`false`。
- **http.proxyHost :** 定义 HTTP 代理的主机名。默认值：`null`。
- **http.proxyPort :** 定义 HTTP 代理的主机端口（整数）。默认值：`null`。
- **http.proxyUser :** 定义 HTTP 代理的用户名。
- **HTTP.proxyPassword :** 定义 HTTP 代理的用户密码。
- **org.uberfire.nio.git.http.enabled:** Enables 或 disable the HTTP 守护进程。默认值：`true`。

- **org.uberfire.nio.git.http.host**: 如果启用了 HTTP 守护进程，它将使用此属性作为主机标识符。这是一个信息性属性，用于显示如何通过 HTTP 访问 Git 存储库。HTTP 仍然依赖于 servlet 容器。默认值：`localhost`。
- **org.uberfire.nio.git.http.hostname**: 如果 HTTP 守护进程被启用，它将使用此属性作为主机名标识符。这是一个信息性属性，用于显示如何通过 HTTP 访问 Git 存储库。HTTP 仍然依赖于 servlet 容器。默认值：`localhost`。
- **org.uberfire.nio.git.http.port** : 如果启用了 HTTP 守护进程，它将使用此属性作为端口号。这是一个信息性属性，用于显示如何通过 HTTP 访问 Git 存储库。HTTP 仍然依赖于 servlet 容器。默认值：`8080`。

## Git over HTTPS

使用以下属性通过 HTTPS 配置对 Git 存储库的访问：

- **org.uberfire.nio.git.proxy.ssh.over.https**: 指定 SSH 是否使用了 HTTPS 代理。默认值：`false`。
- **HTTPS.proxyHost** : 定义 HTTPS 代理的主机名。默认值：`null`。
- **HTTPS.proxyPort** : 定义 HTTPS 代理的主机端口（整数值）。默认值：`null`。
- **HTTPS.proxyUser** : 定义 HTTPS 代理的用户名。
- **HTTPS.proxyPassword** : 定义 HTTPS 代理的用户密码。
- **user.dir** : 用户目录的路径。
- **org.uberfire.nio.git.https.enabled**: 启用或禁用 HTTPS 守护进程。默认值：`false`
- **org.uberfire.nio.git.https.host**: 如果 HTTPS 守护进程已启用，它将使用此属性作为主

机标识符。这是一个信息性属性，用于显示如何通过 HTTPS 访问 Git 存储库。HTTPS 仍然依赖于 servlet 容器。默认值：localhost.

- **org.uberfire.nio.git.https.hostname:** 如果 HTTPS 守护进程已启用，它将使用此属性作为主机名标识符。这是一个信息性属性，用于显示如何通过 HTTPS 访问 Git 存储库。HTTPS 仍然依赖于 servlet 容器。默认值：localhost.
- **org.uberfire.nio.git.https.port:** 如果 HTTPS 守护进程已启用，它将使用此属性作为端口号。这是一个信息性属性，用于显示如何通过 HTTPS 访问 Git 存储库。HTTPS 仍然依赖于 servlet 容器。默认值：8080.

## JGit

- **org.uberfire.nio.jgit.cache.instances** : 定义 JGit 缓存大小。
- **org.uberfire.nio.jgit.cache.overflow.cleanup.size:** 定义 JGit 缓存溢出清理大小。
- **org.uberfire.nio.jgit.remove.eldest.iterations:** Enables 或 disable whether remove eldest JGit 迭代。
- **org.uberfire.nio.jgit.cache.evict.threshold.duration:** 定义 JGit 驱除阈值持续时间。
- **org.uberfire.nio.jgit.cache.evict.threshold.time.unit** : 定义 JGit 驱除阈值单元。

## Git 守护进程

使用以下属性来启用和配置 Git 守护进程：

- **org.uberfire.nio.git.daemon.enabled:** Enables 或 disable the Git 守护进程。默认值：true.
- **org.uberfire.nio.git.daemon.host:** 如果 Git 守护进程已启用，它将使用此属性作为本地主机标识符。默认值：localhost.
- **org.uberfire.nio.git.daemon.hostname:** 如果 Git 守护进程已启用，它将使用此属性作



为本地主机名标识符。默认值：`localhost`

- **org.uberfire.nio.git.daemon.port:** 如果 Git 守护进程已启用，它将使用此属性作为端口号。默认值：`9418`。
- **org.uberfire.nio.git.http.sslVerify:** 启用或禁用 Git 存储库的 SSL 证书检查。默认值：`true`。



#### 注意

如果已经使用默认或者分配的端口，则会自动选择一个新端口。确保端口可用，并检查日志以了解更多信息。

## Git SSH

使用以下属性来启用和配置 Git SSH 守护进程：

- **org.uberfire.nio.git.ssh.enabled:** 启用或禁用 SSH 守护进程。默认值：`true`。
- **org.uberfire.nio.git.ssh.host:** 如果启用了 SSH 守护进程，它将使用此属性作为本地主机标识符。默认值：`localhost`。
- **org.uberfire.nio.git.ssh.hostname:** 如果启用了 SSH 守护进程，它将使用此属性作为本地主机名标识符。默认值：`localhost`。
- **org.uberfire.nio.git.ssh.port:** 如果启用了 SSH 守护进程，它将使用此属性作为端口号。默认值：`8001`。



#### 注意

如果已经使用默认或者分配的端口，则会自动选择一个新端口。确保端口可用，并检查日志以了解更多信息。

- **org.uberfire.nio.git.ssh.cert.dir:** 存储本地证书的 `.security` 目录的位置。默认值：`工作目录`。

- **org.uberfire.nio.git.ssh.idle.timeout:**设置 SSH 闲置超时。
- **org.uberfire.nio.git.ssh.passphrase** : 在使用 SCP 风格 URL 克隆 git 存储库时用于访问操作系统的公钥存储的密码短语。示例：`git@github.com:user/repository.git`。
- **org.uberfire.nio.git.ssh.algorithm:** Algorithm 供 SSH 使用。默认值：`RSA`。
- **org.uberfire.nio.git.gc.limit:** Sets the GC 限值。
- **org.uberfire.nio.git.ssh.ciphers** : 以逗号分隔的密码字符串。可用的密码是 `aes128-ctr,aes192-ctr,aes256-ctr,arcfour128,arcfour256,aes192-cbc,aes256-cbc`。如果没有使用 属性，则会载入所有可用的密码。
- **org.uberfire.nio.git.ssh.macs:**以逗号分隔的消息验证代码(MAC)字符串。可用的 MACs 是 `hmac-md5,hmac-md5-96,hmac-sha1,hmac-sha1-96,hmac-sha2-256,hmac-sha2-512`。如果没有使用 属性，则会加载所有可用的 MAC。



#### 注意

如果您计划使用 RSA 或 DSA 以外的任何算法，请确保将应用服务器设置为使用 Bouncy Castle JCE 库。

## KIE 服务器节点和流程自动化管理器控制器

使用以下属性，从流程自动化管理器控制器配置与 KIE 服务器节点的连接：

- **org.kie.server.controller:** URL 用于连接到 Process Automation Manager 控制器。例如，`ws://localhost:8080/business-central/websocket/controller`。
- **org.kie.server.user:**用于从流程自动化管理器控制器连接到 KIE 服务器节点的用户名。只有在将这个 Business Central 安装用作流程自动化管理器控制器时才需要此属性。
- **org.kie.server.pwd** : 用于从流程自动化管理器控制器连接到 KIE 服务器节点的

**Password**。只有在将这个 **Business Central** 安装用作流程自动化管理器控制器时才需要此属性。

## Maven 和 miscellaneous

使用以下属性配置 **Maven** 和其他其它功能：

- **kie.maven.offline.force**：强制 **Maven** 的行为如离线一样。如果为 **true**，禁用在线依赖关系解析。默认值：**false**。



### 注意

仅在 **Business Central** 中使用此属性。如果您与任何其他组件共享一个运行时环境，请隔离配置并将其应用到 **Business Central**。

- **org.uberfire.gzip.enable**: Enables 或在 **GzipFilter** 压缩过滤器中禁用 **Gzip** 压缩。默认值：**true**。
- **org.kie.workbench.profile**: 选择 **Business Central** 配置集。可能的值有 **FULL** 或 **PLANNER\_AND\_RULES**。一个前缀 **FULL\_** 设定配置集，并从管理员首选项中隐藏配置集首选项。默认值：**FULL**
- **org.appformer.m2repo.url**: **Business Central** 在查找依赖项时使用 **Maven** 存储库的默认位置。它定向到 **Business Central** 中的 **Maven** 存储库，例如 **http://localhost:8080/business-central/maven2**。在启动 **Business Central** 前设置此属性。默认值：到内 **m2** 存储库的文件路径。
- **appformer.ssh.keystore**：通过指定类名称定义要与 **Business Central** 搭配使用的自定义 **SSH** 密钥存储。如果属性不可用，则使用默认的 **SSH** 密钥存储。
- **appformer.ssh.keys.storage.folder**：在使用默认 **SSH** 密钥存储时，此属性为用户的 **SSH** 公钥定义存储文件夹。如果属性不可用，则密钥将存储在 **Business Central .security** 文件夹中。
- **appformer.experimental.features**: 启用实验功能框架。默认值：**false**。

- **org.kie.demo** : 启用来自 **GitHub** 的演示应用程序的外部克隆。
- **org.uberfire.metadata.index.dir** : 存储 **Lucene .index** 目录的位置。默认值 : 工作目录。
- **org.uberfire ldap.regex.role\_mapper**: **Regex** 模式用于将 **LDAP** 主体名称映射到应用程序角色名称。请注意, 在匹配原则值和角色名称时, 变量角色必须是模式的一部分, 因为应用程序角色名称替换变量角色。
- **org.uberfire.sys.repo.monitor.disabled** : 禁用配置监控器。除非确定, 否则不要禁用。默认值 : **false**。
- **org.uberfire.secure.key** : 密码加密使用的密码。默认值 : **org.uberfire.admin**。
- **org.uberfire.secure.alg**: **Crypto** 算法由密码加密使用。默认值 : **PBEWithMD5AndDES**。
- **org.uberfire.domain**: **uberfire** 使用的 **Security-domain name**。默认值 : **ApplicationRealm**。
- **org.guvnor.m2repo.dir**: **Place** 存储 **Maven** 存储库文件夹。默认值 : **&lt ;working-directory>/repositories/kie**。
- **org.guvnor.project.gav.check.disabled**: **Disables group ID、工件 ID 和版本(GAV)**检查。默认值 : **false**。
- **org.kie.build.disable-project-explorer** : 禁用 **Project Explorer** 中所选项目的自动构建。默认值 : **false**。
- **org.kie.builder.cache.size** : 定义项目构建器的缓存大小。默认值 : **20**。
- **org.kie.library.assets\_per\_page**:您可以在项目屏幕中自定义每个页面的资产数量。默认值 : **15**。

- **org.kie.verification.disable-dtable-realtime-verification:** 禁用决策表的实时验证和验证。默认值：**false**。

## 进程自动化管理器控制器

使用以下属性配置如何连接到流程自动化管理器控制器：

- **org.kie.workbench.controller:** 用于连接 Process Automation Manager 控制器的 URL，例如 `ws://localhost:8080/kie-server-controller/websocket/controller`。
- **org.kie.workbench.controller.user:** Process Automation Manager 控制器用户。默认值：**kieserver**。
- **org.kie.workbench.controller.pwd:** Process Automation Manager 控制器密码。默认值：**kieserver1!**。
- **org.kie.workbench.controller.token:** 用于连接到 Process Automation Manager 控制器的令牌字符串。

## Java Cryptography Extension KeyStore(JCEKS)

使用以下属性配置 JCEKS：

- **kie.keystore.keyStoreURL**：用于加载 Java Cryptography Extension KeyStore(JCEKS)的 URL。例如，`file:///home/kie/keystores/keystore.jceks`。
- **kie.keystore.keyStorePwd**：用于 JCEKS 的密码。
- **kie.keystore.key.ctrl.alias:**默认 REST Process Automation Manager 控制器的密钥别名。
- **kie.keystore.key.ctrl.pwd:**默认 REST Process Automation Manager 控制器的别名。

渲染

使用以下属性在 **Business Central** 和 **KIE 服务器** 呈现的形式间切换：

- **org.jbpm.wb.forms.renderer.ext** : 切换 **Business Central** 和 **KIE 服务器** 之间的表单渲染。默认情况下，表单渲染由 **Business Central** 执行。默认值：**false**。
- **org.jbpm.wb.forms.renderer.name**: 让您在 **Business Central** 和 **KIE 服务器** 呈现的表单之间进行切换。默认值：**workbench**。

## 部分 V. 在 IBM WEBSHERE APPLICATION SERVER 上安装和配置 KIE 服务器

作为系统管理员，您可以为红帽 KIE 服务器配置 IBM WebSphere Application Server，并在 IBM 服务器实例上安装 KIE 服务器。



### 注意

现在，在维护阶段，对 IBM WebSphere Application Server 上的 Red Hat Process Automation Manager 的支持。红帽将继续在 IBM WebSphere Application Server 上支持 Red Hat Process Automation Manager，但有以下限制：

- 红帽不会发布新的认证或软件功能。
- 红帽只会发布具有重大影响和关键任务程序错误修复补丁的合格安全修补程序。

未来，红帽可能会将客户定向到与红帽混合云策略兼容的新平台和产品组件。

### 先决条件

- 安装了 IBM WebSphere Application Server 实例版本 9.0 或更高版本。有关完整安装说明，请参阅 [IBM WebSphere Application Server 产品页面](#)。
- 您可以访问 WebSphere 集成解决方案控制台，通常位于 <http://<HOST>:9060/ibm/console>。

## 第 44 章 KIE 服务器

**KIE 服务器是保存和运行红帽流程自动化管理器的规则和其他工件的服务器。KIE 服务器是一个独立内置组件，可用于通过可用于 REST、Java 消息服务(JMS)或 Java 客户端平台应用程序的接口实例化和执行规则，以及通过解决者管理进程、作业和红帽构建 OptaPlanner 功能。**

**创建为 Web 可部署 WAR 文件，可以在任何 web 容器中部署 KIE 服务器。当前版本的 KIE 服务器包含在 Red Hat Decision Manager 和 Red Hat Process Automation Manager 的默认扩展中。**

**KIE 服务器占用内存占用量较低，因此可在云中轻松部署。此服务器的每个实例都可以打开并实例化多个容器，可让您并行执行多个规则服务。**

**KIE 服务器可以与其他应用服务器（如 Oracle WebLogic Server 或 IBM WebSphere Application Server）集成，以简化红帽流程自动化管理器应用程序管理。**



## 第 45 章 IBM WEBSHERE APPLICATION SERVER

**IBM WebSphere Application Server 是一个灵活、安全的 Web 应用程序服务器，托管基于 Java 的 Web 应用程序，并提供 Java EE 认证的运行时环境。IBM WebSphere 9.0 支持 Java SE 8，且完全兼容 Java EE 7。**

## 第 46 章 安装和运行 IBM WEBSHERE APPLICATION SERVER

必须安装并运行 **IBM WebSphere Application Server**，才能应用适用于 KIE 服务器的许多配置。这部分论述了如何安装和启动 **IBM WebSphere**。

有关最新和详细的安装说明，请参阅 [IBM 知识库](#)。

### 流程

1. 从 [IBM 安装 管理器和打包实用程序下载链接页面](#) 下载 **IBM 安装管理器** 版本 1.8.5 或更高版本。安装 **IBM WebSphere** 需要 **IBM 安装管理器**。

2. 解压下载的存档，并以 **root** 用户身份在新目录中运行以下命令：

```
sudo ./install
```

此时会打开 **IBM 安装管理器**。

3. 前往 **File** → **preferences**，然后单击 **Add Repository**。
4. 在 **Add Repository** 窗口中，输入 **IBM WebSphere 9.0** 的存储库 URL。您可以在 [IBM 知识库的 IBM WebSphere Application Server 产品页面中找到在线](#) 产品存储库中的所有存储库 URL。
5. 在命令终端中，导航到在安装过程中指定的 **IBM WebSphere Application Server** 文件夹位置。
6. 切换到 **/bin** 目录并运行类似以下示例的命令，以创建 **IBM WebSphere** 配置集、用户名和密码。配置集定义运行时环境。该配置集包括服务器在运行时环境中进程以及您可以更改的所有文件。登录需要该用户。

```
sudo ./manageprofiles.sh -create -profileName testprofile -profilePath /profiles/testprofile -adminUserName websphere -adminPassword password123
```

7. 在命令终端中，导航到您创建的配置集中的 **bin** 目录（例如 **/profiles/testprofile/bin**），再运行以下命令来启动 **IBM WebSphere Application Server** 实例：

```
sudo ./startServer.sh <SERVER_NAME>
```

将 **<SERVER\_NAME >** 替换为在 **Servers → Server Types → IBM WebSphere Application Server of the WebSphere Integrated Solutions Console** 中定义的 **IBM WebSphere Application Server** 名称。

8.

在网页浏览器中打开以下 URL :

```
http://<HOST>:9060/ibm/console
```

**<HOST >** 是目标服务器的系统名称或 IP 地址。

例如，要为在系统中运行的 **IBM WebSphere** 实例启动 **WebSphere Integrated Solutions** 控制台，在网页浏览器中输入以下 URL :

```
http://localhost:9060/ibm/console
```

9.

当显示 **WebSphere 集成解决方案** 控制台的登录页面时，请输入您的管理凭证。

## 第 47 章 为 KIE 服务器配置 IBM WEBSHERE APPLICATION SERVER

在使用 IBM WebSphere 应用服务器部署 KIE 服务器前，您必须在 IBM WebSphere 上配置系统属性、安全设置、JMS 要求和其他属性。这些配置促进与 KIE 服务器的最佳集成。

### 先决条件

- IBM WebSphere 应用服务器已安装并运行。
- 已登陆到 WebSphere Integrated Solutions 控制台。

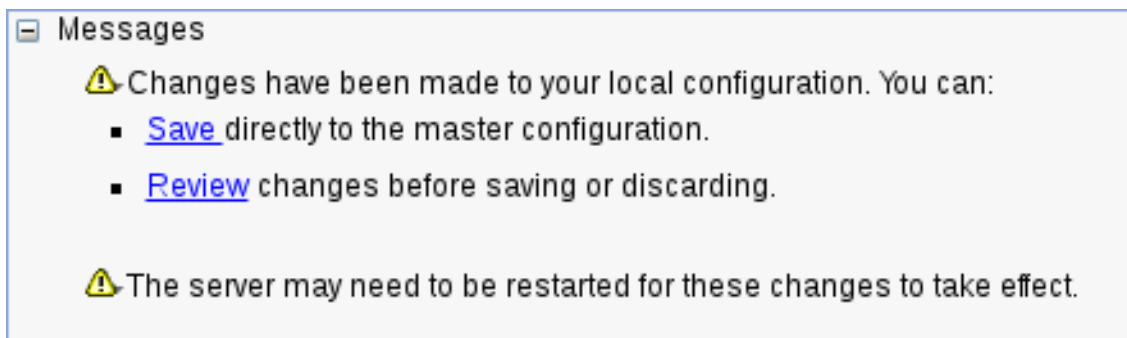
### 47.1. 启用管理安全性

您必须在 WebSphere Integrated Solutions 控制台中启用管理安全性，以便您有创建用户和组所需的权限。

### 流程

1. 在 WebSphere Integrated Solutions Console 中，点 **Security** → **Global Security**，并确保选择了 **Enable Application Security** 选项。可能已经选择它并在服务器级别上覆盖。
2. 点击 **Security Configuration Wizard** 并点 **Next**。
3. 选择包含用户信息的存储库。例如，为本地配置选择 **Federated** 软件仓库。
4. 点 **Next**
5. 输入主管理用户名和密码。
6. 单击 **Next**，然后单击 **Finish**。
7. 点 **Messages** 窗口中的 **Save**，将您的更改保存到主配置中。

图 47.1. 保存安全更改



8.

在命令终端中，进入在安装过程中指定的 IBM WebSphere Application Server /bin 目录位置，并运行以下命令来停止和重启 IBM WebSphere 以应用安全更改：

```
sudo ./stopServer.sh <SERVER_NAME>
```

```
sudo ./startServer.sh <SERVER_NAME>
```

将 `<SERVER_NAME >` 替换为在 **Servers** → **Server Types** → **IBM WebSphere Application Server of the WebSphere Integrated Solutions Console** 中定义的 **IBM WebSphere Application Server** 名称。

## 47.2. 在 IBM WEBSHERE APPLICATION SERVER 中配置 JDBC 数据源

数据源是一个对象，它允许 Java 数据库连接(JDBC)客户端（如应用服务器）与数据库建立连接。应用程序在 Java 命名和目录接口(JNDI)树或本地应用程序上下文中查找数据源，并请求数据库连接来检索数据。您必须为 IBM WebSphere Application Server 配置数据源，以确保在服务器和指定数据库之间进行正确的数据交换。

通常，使用 Red Hat Process Automation Manager 的解决方案在单个事务中管理多个资源。用于异步作业、事件和计时器的 JMS，例如：在确保数据原子性和一致性结果时，Red Hat Process Automation Manager 需要数据源中的 XA 驱动程序。如果监听器内存在针对不同架构的事务代码，或者从 jBPM 引擎提供的 hook 派生出来，则还需要一个 XA 驱动程序。

不要使用非 XA 数据源，除非您正面没有参与单一事务的多个资源。

### 先决条件

•

您要用于创建数据库连接的 JDBC 供应商会在您要部署数据源的所有服务器上配置。有关 JDBC 供应商的更多信息，请参阅 IBM 知识库中的 [配置 JDBC 供应商](#)。

## 流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：

- 产品：流程自动化管理器
- Version: 7.13.5

2. 下载 Red Hat Process Automation Manager 7.13.5 附加组件。

3. 完成以下步骤以准备数据库：

- a. 在临时目录中提取 `rhpmam-7.13.5-add-ons.zip`，如 `TEMP_DIR`。
- b. 提取 `TEMP_DIR/rhpmam-7.13.5-migration-tool.zip`。
- c. 将您的当前目录更改为 `TEMP_DIR/rhpmam-7.13.5-migration-tool/ddl-scripts` 目录。此目录包含多种数据库类型的 DDL 脚本。
- d. 将您的数据库类型的 DDL 脚本导入到您要使用的数据库中，例如：

```
psql jbpm < /ddl-scripts/postgresql/postgresql-jbpm-schema.sql
```



### 注意

如果您与 Spring Boot 结合使用 PostgreSQL 或 Oracle，则必须导入对应的 Spring Boot DDL 脚本，如 `/ddl-scripts/oracle/oracle-springboot-jbpm-schema.sql` 或 `/ddl-scripts/postgresql/postgresql-springboot-jbpm-schema.sql`。

4. 在 WebSphere 集成解决方案控制台中，导航到 `Resources` → `JDBC` → `Data sources`。

5. 选择应用程序可以使用数据源的范围。您可以选择 **cell**、**node**、**cluster** 或 **server**。
6. 点击 **New** 打开 **Create a data source** 向导。
7. 在 **Data source name** 字段中输入唯一名称，在 **JNDI name** 字段中输入 **Java Naming and Directory Interface(JNDI)**名称。应用程序服务器使用 **JNDI** 名称，可以将应用的资源引用绑定到此数据源。

不要在不同的资源类型间分配重复的 **JNDI** 名称，如数据源和 **J2C** 连接工厂或 **JMS** 连接工厂。

不要为同一范围内相同类型的多个资源分配重复的 **JNDI** 名称。
8. 如果创建了提供程序，请单击 **Next**，然后选择 **Select an existing JDBC provider**，或者选择 **Create new JDBC provider** 来为新提供程序定义详情。（建议此点前调整 **JDBC** 提供程序，作为添加数据源的先决条件。）
9. 在数据源面板的 **Enter 数据库特定属性** 上，如果容器管理的持久性(**CMP**)**enterprise Bean** 必须访问此数据源，请单击容器管理的持久性(**CMP**)中使用此数据源。然后，填写任何其他数据库特定属性。
10. 可选：配置数据源的安全别名。您可以为任何身份验证方法选择 **None**，或者选择以下类型之一：
  - 组件管理的验证别名：指定当组件资源引用 **res-auth** 值是 **Application** 时使用的身份验证别名。要定义新的别名，请导航到 **相关 Items** → **J2EE 连接器架构(J2C)**身份验证数据条目。组件管理的别名代表在应用程序中指定的用于数据源身份验证的 **ID** 和密码的组合。因此，您在数据源上设置的别名必须与应用程序代码中的别名相同。
  - **mapping-configuration** 别名：仅在组件资源引用中没有登录配置时使用。对组件资源引用上的登录配置和关联属性的规格是在将 **res-auth** 值设置为 **Container** 时定义身份验证策略的首选方法。如果您指定了 **DefaultPrincipalMapping** 登录配置，相关的属性是一个 **JAAS - J2C** 身份验证数据条目别名。
  - 容器管理的验证别名：仅在组件资源引用中没有登录配置时才使用。当 **res-auth** 值设置为 **Container** 时，组件资源引用上的登录配置和关联属性决定了容器管理的验证策略。

11.

点 **Next**，查看数据源的信息，然后点 **Finish** 保存配置并退出向导。

**Data sources** 面板在表中显示您的新配置，以及为相同范围配置的任何其他数据源。

有关 IBM WebSphere 应用服务器数据源的更多信息，请参阅 IBM 知识库中的 [配置 JDBC 供应商和数据源](#)。

### 47.3. 配置 JAVA 消息服务(JMS)

Java 消息服务(JMS)是一个 Java API，KIE 服务器用来与其他应用服务器（如 Oracle WebLogic Server 和 IBM WebSphere Application Server）交换消息。您必须将应用服务器配置为通过 KIE 服务器发送和接收 JMS 消息，以确保两个服务器之间的协作。

#### 47.3.1. 创建服务总线并添加 IBM WebSphere Application Server

您必须创建一个服务总线，并添加 IBM WebSphere Application Server 作为成员才能使用 JMS。

#### 流程

1. 在 WebSphere 集成解决方案控制台中，导航到 **Service Integration** → **Buses** → **New**。
2. 输入新总线名称并清除 **总线安全** 选项。
3. 单击 **Next**，然后单击 **Finish** 以创建服务总线。
4. 选择您创建的服务总线。
5. 在 **Topology** 中，选择 **Bus Members** → **Add**。
6. 在 **Add a New Bus Member** 向导中，选择 **IBM WebSphere Application Server** 和消息存储以实现持久性。您还可以指定消息存储的属性。



7. 点 **Finish** 添加新总线成员。

### 47.3.2. 创建 JMS 连接工厂

要使用 KIE 服务器启用消息，您必须创建某些 JMS 连接工厂以发送和接收消息。

#### 先决条件

- 您已为 IBM WebSphere Application Server 创建服务总线。

#### 流程

1. 在 WebSphere 集成解决方案控制台中，导航到 **Resources** → **JMS** → **Connection Factories**。
2. 选择正确的范围，再单击 **New**。
3. 选择 **Default Messaging Provider** 选项，再单击 **OK**。
4. 对于以下每个所需连接工厂，请输入连接工厂的名称（如 **KIE.SERVER.REQUEST**）和 **JNDI** 名称（例如：**jms/cf/KIE.SERVER.REQUEST**），然后从 **Bus Name** 下拉列表中选择服务总线。保留其余选项的默认值。
5. 点 **Apply**，然后点 **Save** 将更改保存到主配置中，然后对每个所需工厂重复这个更改。

#### 47.3.2.1. KIE 服务器的 JMS 连接工厂

下表列出了启用使用 KIE 服务器的 JMS 消息服务(JMS)连接工厂所需的 Java 消息服务(JMS)连接：

表 47.1. KIE 服务器所需的 JMS 连接工厂

Name	默认值	用于
<b>KIE.SERVER.REQUEST</b>	<b>jms/cf/KIE.SERVER.REQUEST</b>	将所有请求发送到 KIE 服务器

Name	默认值	用于
<b>KIE.SERVER.RESPONSE</b>	<b>jms/cf/KIE.SERVER.RESPONSE</b>	接收 KIE 服务器生成的所有响应
<b>KIE.SERVER.EXECUTOR</b>	<b>jms/cf/KIE.SERVER.EXECUTOR</b>	KIE 服务器执行程序服务

### 47.3.3. 创建 JMS 队列

**JMS 队列**是点对点消息传递的目标端点。您必须创建特定的 **JMS 队列**，以启用使用 **KIE 服务器**的 **JMS 消息传递**。

#### 先决条件

- 您已为 **IBM WebSphere Application Server** 创建服务总线。

#### 流程

1. 在 **WebSphere Integrated Solutions Console** 中，导航到 **Resources** → **JMS** → **Queues**。
2. 选择正确的范围，再单击 **New**。
3. 选择 **Default Messaging Provider** 选项，再单击 **OK**。
4. 对于以下每个所需队列，输入队列的名称（如 **KIE.SERVER.REQUEST**）和 **JNDI 名称**（如 **jms/KIE.SERVER.REQUEST**），然后从 **bus Name** 下拉列表中选择服务总线。
5. 从 **Queue Name** 下拉列表中，选择 **Create Service Integration Bus Destination**，输入唯一标识符，然后选择您之前创建的总线成员。
6. 单击 **Apply**，然后单击 **Save**，将更改保存到主配置中，然后重复对每个所需的队列。

#### 47.3.3.1. KIE 服务器的 JMS 队列

下表列出了使用 KIE 服务器启用 JMS 消息传递所需的 Java 消息服务(JMS)队列：

表 47.2. KIE 服务器所需的 JMS 队列

Name	默认值	用于
KIE.SERVER.REQUEST	jms/KIE.SERVER.REQUEST	将所有请求发送到 KIE 服务器
KIE.SERVER.RESPONSE	JMS/KIE.SERVER.RESPONSE	接收 KIE 服务器生成的所有响应
KIE.SERVER.EXECUTOR	JMS/KIE.SERVER.EXECUTOR	KIE 服务器执行程序服务

#### 47.3.4. 创建 JMS 激活规格

需要 JMS 激活规格来桥接队列，以及启用 JMS 的消息驱动的 Bean。

##### 先决条件

- 您已为 IBM WebSphere Application Server 创建服务总线。
- 您已创建了 JMS 队列。

##### 流程

1. 在 WebSphere 集成解决方案控制台中，导航到 **Resources** → **JMS** → **Activation Specifications**。
2. 选择正确的范围，再单击 **New**。
3. 选择 **Default Messaging Provider** 选项，再单击 **OK**。
4. 对于以下每个需要的激活规格，输入激活规格的名称（如 **KIE.SERVER.REQUEST**）和 JNDI 名称（例如，**jms/activation/KIE.SERVER.REQUEST**），然后从 **Bus Name** 下拉列表中选择服务总线。

5. 从 **Destination Type** 下拉列表中，选择 **Queue**，并将对应的队列的名称输入为 **Destination** 查找（例如，`jms/KIE.SERVER.REQUEST`）。
6. 单击 **Apply**，然后单击 **Save**，将更改保存到主配置中，再重复每个所需的激活规格。

#### 47.3.4.1. KIE 服务器的 JMS 激活规格

下表列出了启用使用 KIE 服务器的 JMS 消息传递所需的 Java 消息服务(JMS)激活规格：

表 47.3. KIE 服务器所需的 JMS 激活规格

Name	默认值	用于
<b>KIE.SERVER.REQUEST</b>	<code>jms/activation/KIE.SERVER.REQUEST</code>	将所有请求发送到 KIE 服务器
<b>KIE.SERVER.RESPONSE</b>	<code>JMS/activation/KIE.SERVER.RESPONSE</code>	接收 KIE 服务器生成的所有响应
<b>KIE.SERVER.EXECUTOR</b>	<code>JMS/activation/KIE.SERVER.EXECUTOR</code>	KIE 服务器执行程序服务

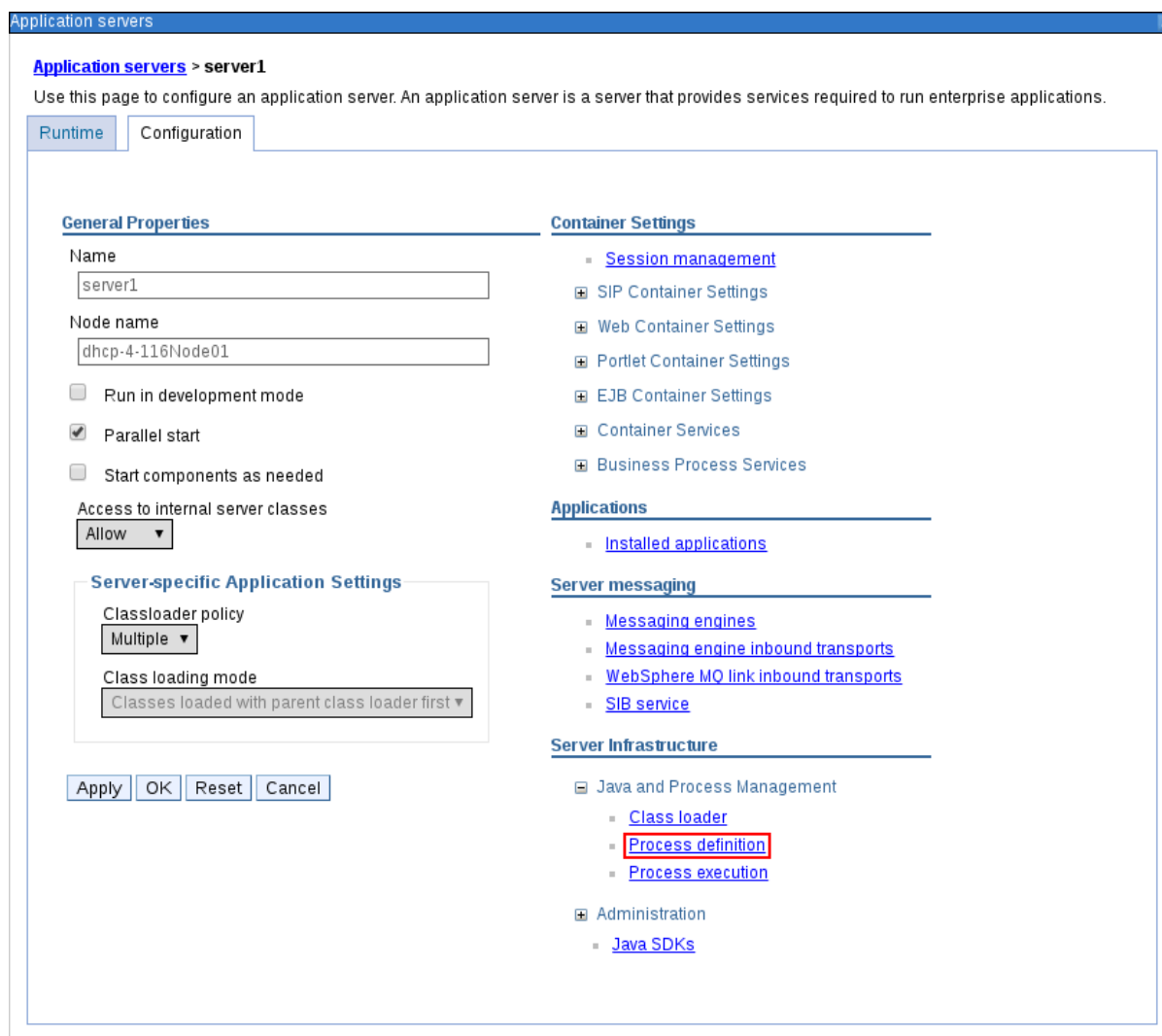
#### 47.4. 在 IBM WEBSHERE APPLICATION SERVER 中设置系统属性

在部署 KIE 服务器前，设置 IBM WebSphere Application Server 中列示的系统属性。

##### 流程

1. 在 WebSphere Integrated Solutions Console 中，导航到 **Servers** → **Server Types** → **IBM WebSphere Application Servers**。
2. 在应用服务器列表中，选择要在其上部署 KIE 服务器的服务器。
3. 在 **Server Infrastructure** 下，点 **Java and Process Management** → **Process Definition**。

图 47.2. WebSphere 配置页面



4.

在 **Additional Properties** 下，点 **Java Virtual Machine**。

图 47.3. 进程定义配置页面

Application servers > server1 > Process definition

Use this page to configure a process definition. A process definition defines the command line information necessary to start or initialize a process.

Configuration

General Properties	Additional Properties
Executable name <input type="text"/>	<ul style="list-style-type: none"> <li>Java Virtual Machine</li> <li>Environment Entries</li> <li>Process execution</li> <li>Process Logs</li> <li>Logging and tracing</li> </ul>
Executable arguments <input type="text"/>	
Start command <input type="text"/>	
Start command arguments <input type="text"/>	
Stop command <input type="text"/>	
Stop command arguments <input type="text"/>	
Working directory <input type="text" value="\${USER_INSTALL_ROOT}"/>	
Executable target type JAVA_CLASS	
Executable target <input type="text" value="com.ibm.ws.runtime.WsServer"/>	

Apply OK Reset Cancel

这会打开用于启动 IBM WebSphere 的 JVM 的配置属性。

5. 将 Initial 堆大小 和 Maximum 堆大小设为 2048，再点击 Apply 来提高 Java 虚拟机(JVM) 内存大小。KIE 服务器已使用这些值进行了测试。如果您没有增加 JVM 内存大小，IBM WebSphere Application Server freezes 或在部署 KIE 服务器时导致部署错误。
6. 在 "其他属性 "下，单击 "自定义属性"。
7. 点 New → Custom JVM Properties，并在 IBM WebSphere 中添加以下属性：

表 47.4. KIE 服务器的系统属性

Name	值	描述
<b>kie.server.jms.queues.response</b>	<b>JMS/KIE.SERVER.RESPONSE</b>	用于 KIE 服务器使用的响应的 JMS 队列的 JNDI 名称。
<b>org.kie.server.domain</b>	<b>WSLogin</b>	在使用 JMS 时用于验证用户的身份的 JAAS <b>LoginContext</b> 域。
<b>org.kie.server.persistence.ds</b>	<b>jdbc/jbpm</b>	KIE 服务器的数据源 JNDI 名称。
<b>org.kie.server.persistence.tm</b>	<b>org.hibernate.engine.transaction.jta.platform.internal.WebSphereJtaPlatform</b>	用于设置 Hibernate 属性的事务管理器平台。
<b>org.kie.server.persistence.dialect</b>	示例： <b>org.hibernate.dialect.H2Dialect</b>	指定要使用的 Hibernate 权利。根据数据源设置。
<b>org.kie.executor.jms.queue</b>	<b>JMS/KIE.SERVER.EXECUTOR</b>	KIE 服务器的作业执行器 JMS 队列。
<b>org.kie.executor.jms.cf</b>	<b>jms/cf/KIE.SERVER.EXECUTOR</b>	KIE 服务器的作业执行器 JMS 连接工厂。
<b>org.kie.server.router</b>	示例： <b>http://localhost:9000</b>	(可选) 指定应用服务器在集群 KIE 服务器环境中包含一个或多个 KIE 服务器路由器(Smart Router)的一个或多个 URL。
<b>org.jboss.logging.provider</b>	<b>jdk</b>	只有环境中安装了 <b>CA SiteMinder TAI(SMTAI)</b> 时，才需要此属性。使用此属性会强制 Hibernate 使用 <b>JDK</b> ，而不是 <b>log4j</b> 在 Dashbuilder 中登录。 <b>CA siteMinder TAI(SMTAI)</b> 包含 <b>log4j</b> 的旧版本，从而导致冲突。

8.

点 Save 保存对主配置的更改。

#### 47.5. 停止并重启 IBM WEBSHERE APPLICATION SERVER

在 IBM WebSphere Application Server 中配置了所有所需的系统属性后，停止并重启 IBM 服务器，以确保应用了配置。

## 流程

在命令终端中，进入在安装过程中指定的 **IBM WebSphere Application Server /bin** 目录位置，并运行以下命令来停止和重启 **IBM WebSphere** 以应用配置更改：

```
sudo ./stopServer.sh <SERVER_NAME>
```

```
sudo ./startServer.sh <SERVER_NAME>
```

将 **<SERVER\_NAME >** 替换为在 **Servers → Server Types → IBM WebSphere Application Server of the WebSphere Integrated Solutions Console** 中定义的 **IBM WebSphere Application Server** 名称。



## 第 48 章 使用 IBM WEBSHERE 应用服务器安装并运行 KIE 服务器

在 IBM WebSphere Application Server 中配置所有所需的系统属性后，您可以使用 IBM WebSphere 安装 KIE 服务器，以简化红帽流程自动化管理器应用程序管理。

### 先决条件

- IBM WebSphere Application Server 实例配置为如 [第 47 章 为 KIE 服务器配置 IBM WebSphere Application Server](#) 所述。

### 流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
  - 产品：流程自动化管理器
  - Version: 7.13.5
2. 下载 适用于所有支持的 EE7 容器的 Red Hat Process Automation Manager 7.13.5 KIE Server。
3. 将 `rhpm-7.13.5-kie-server-ee7.zip` 存档提取到临时目录中。在以下示例中，此目录名为 `TEMP_DIR`。
4. 重新打包 `kie-server.war` 目录：
  - a. 导航到 `TEMP_DIR/rhpm-7.13.5-kie-server-ee7/kie-server.war` 目录。
  - b. 选择 `TEMP_DIR/rhpm-7.13.5-kie-server-ee7/kie-server.war` 目录的内容，再创建 `kie-server.zip` 文件。
  - c. 将 `kie-server.zip` 重命名为 `kie-server.war`。这是您要用来部署 KIE 服务器的文件。

- d. 可选：将新的 `kie-server.war` 文件复制到更方便地从中进行部署的位置。
5. 在 WebSphere Integrated Solutions 控制台中，导航到 **Applications** → **Application Types** → **WebSphere Enterprise Applications**。
6. 点 **InstCll**。
7. 导航到您重新打包的 `kie-server.war` 文件，并选择其上传。
8. 选择 **Fast Path**，再单击 **Next**。  
  
此时会打开 **Install New Application** 向导。
9. 将 **Application Name** 更改为 `kie-server`，然后单击 **Next**。
10. 根据您的特定要求将 **KIE 服务器模块** 映射到服务器，然后单击下一步。
11. 对于 **Message-Driven Beans** 的 **Bind Listeners**，为两个 **Beans** 选择 **Activation Specification**，在 **Target Resource JNDI Name** 字段中输入 `.jms/activation/KIE.SERVER.REQUEST`。
12. 在 **Map Virtual Hosts for Web Modules** 部分中，保留默认值，然后单击 **Next**。
13. 将上下文 **root** 设置为 `kie-server`。
14. 在 **模块的元数据** 部分中，保留默认值，然后单击下一步。
15. 单击 **Finish** 以安装 **KIE 服务器**，然后单击 **Save** 以将更改保存到主配置中。

#### 48.1. 创建 KIE 服务器组和角色

安装 KIE 服务器后，您必须创建 **kie-server** 组 and 用户。

#### 先决条件

- KIE 服务器安装在 IBM WebSphere Application Server 实例中。

#### 流程

1. 在 WebSphere 集成解决方案控制台中，点击 **用户和组群** → **Manage Groups**。
2. 在 **Manage Groups** 屏幕中，单击 **Create**。
3. 在 **Create a Group** 屏幕中，在 **Group name** 框中输入 **kie-server**，然后单击 **Create**。
4. 要创建添加到 **kie-server** 组的用户，点 **Users** 和 **Groups** → **Manage Users**。
5. 在创建用户 屏幕中，填写所需信息。
6. 单击 **Group Membership**。
7. 在 **Group Membership** 屏幕中，单击 **kie-server**，将它移到 **Mapped To**，然后单击 **Close**。
8. 在创建用户 屏幕上，单击 **Create**。

#### 48.2. 映射 KIE 服务器组和角色

安装 KIE 服务器后，您必须将 **kie-server** 角色映射到 WebSphere 集成解决方案控制台中的 **kie-server** 组来运行 KIE 服务器。

#### 先决条件

- **KIE 服务器安装在 IBM WebSphere Application Server 实例中。**
- **IBM WebSphere Application Server 具有至少一个用户的 kie-server 组。**

## 流程

1. **在 WebSphere Integrated Solutions 控制台中，导航到 Applications → Application Types → WebSphere Enterprise Applications，然后选择新安装的 kie-server 应用程序。**
2. **在详细属性下，点 Security Role to User/Group Mapping。**
3. **选择 kie-server 角色并点击 Map Groups 搜索 kie-server 组。**
4. **将 kie-server 组从 Available 列表移到 Selected 列表中，然后单击 OK。**

**此映射可让 IBM WebSphere Application Server kie-server 组中的用户访问 KIE 服务器。**

5. **单击 Save 以完成映射。**

### 48.3. 为 KIE 服务器配置类加载

**安装 KIE 服务器后，您必须配置类加载，以将父类设置为最后加载。**

## 流程

1. **导航到 Applications → Application Types → WebSphere Enterprise Applications，点 kie-server。**
2. **点左侧 Details Properties 标题下的 Class Loading and Update Detection。**
3. **在属性中，将 Class Loader Order 更改为以本地类加载的类（最后一个）和 WAR Class Loader Policy to Single class loader for application。**

4. 保存对主配置的更改。

#### 48.4. 验证安装

安装 KIE 服务器并定义 KIE 服务器组映射后，验证该服务器是否正在运行。

##### 先决条件

- KIE 服务器安装在 IBM WebSphere Application Server 实例中。
- 您已为无头进程自动化管理器控制器设置所有所需的系统属性。
- 您已在 IBM WebSphere Application Server 中定义了 KIE Server 组映射。

##### 流程

1. 要验证服务器是否正在运行，请完成以下任务之一：
  - 进入 KIE 服务器 URL `http://<HOST>:<PORT>/kie-server`。
  - 发送 GET 请求到 `http://<HOST>:<PORT>/kie-server/services/rest/server`，以检查 KIE 服务器 REST API 是否响应。

在这些示例中，替换以下占位符：

    - `<HOST >` 是无头进程 Automation Manager 控制器的 ID 或名称，如 `localhost` 或 `192.7.8.9`。
    - `<PORT >` 是 KIE 服务器主机的端口号，例如 `9060`。
2. 如果 KIE 服务器没有运行，请停止并重启 IBM WebSphere Application Server 实例，并尝试再次尝试访问 KIE 服务器 URL 或 API。

## 第 49 章 使用 IBM WEBSHERE APPLICATION SERVER 安装并运行无头进程自动化管理器控制器

要使用 KIE 服务器 REST API 或 Java Client API 与 KIE 服务器交互，请使用 IBM WebSphere Application Server 安装无头进程自动化管理器控制器。无头进程自动化管理器控制器以集中的方式管理 KIE 服务器配置，以便您可以使用无头进程自动化管理器控制器创建和维护容器并执行其他服务器级任务。



### 注意

要获得最佳结果，请在生产环境中的不同服务器中安装 KIE 服务器和无头进程自动化管理器控制器。在开发环境中，您可以在同一服务器上安装 KIE 服务器和无头进程自动化管理器控制器。

### 先决条件

- IBM WebSphere Application Server 实例配置如 [第 47 章 为 KIE 服务器配置 IBM WebSphere Application Server](#) 所述。
- KIE 服务器安装在 IBM WebSphere Application Server 实例中。
- 有足够的用户权限才能完成安装。

### 流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
  - 产品：流程自动化管理器
  - Version: 7.13.5
2. 下载 Red Hat Process Automation Manager 7.13.5 附加组件。
3. 将下载的 `rhpm-7.13.5-add-ons.zip` 文件提取到临时目录中。

4. 将 `rhpm-7.13.5-add-ons.zip` 文件提取到临时目录中。在以下示例中，此目录名为 `TEMP_DIR`。
5. 提取 `rhpm-7.13.5-add-ons/rhpm-7.13.5-controller-ee7.zip` 文件。
6. 打包 `controller.war` 目录：
  - a. 导航到 `TEMP_DIR/rhpm-7.13.5-add-ons/rhpm-7.13.5-controller-ee7/controller.war` 目录。
  - b. 选择 `TEMP_DIR/rhpm-7.13.5-add-ons/rhpm-7.13.5-controller-ee7/controller.war` 目录的内容，并创建 `controller.zip` 文件。
  - c. 将 `controller.zip` 重命名为 `controller.war`。这是您要用来部署无头进程自动化管理器控制器的文件。
  - d. 可选：将新的 `controller.war` 文件复制到更方便地从中进行部署的位置。
7. 在 WebSphere Integrated Solutions 控制台中，导航到 `Applications → Application Types → WebSphere Enterprise Applications`。
8. 点 `Install`。
9. 导航到您重新打包的 `controller.war` 文件，并选择它上传。
10. 选择 `Fast Path`，再单击 `Next`。

此时会打开 `Install New Application` 向导。
11. 将 `Application Name` 更改为 `controller`，再单击 `Next`。

12. 根据您的特定要求将无外设进程自动化管理器控制器模块映射到服务器，然后点 **Next**。
13. 对于 **Message-Driven Beans** 的 **Bind Listeners**，为两个 **Beans** 选择 **Activation Specification**，在 **Target Resource JNDI Name** 字段中输入 **jms/ activation/ KIE.SERVER.REQUEST**。
14. 在 **Map Virtual Hosts for Web Modules** 部分中，保留默认值，然后单击 **Next**。
15. 将上下文 **root** 设置为 **controller**。
16. 在 **模块的元数据** 部分中，保留默认值，然后单击下一步。
17. 点 **Finish** 安装无头进程 **Automation Manager** 控制器，然后点 **Save** 保存对主配置的更改。

#### 49.1. 为无外进程自动化管理器控制器配置类加载

安装无头进程 **Automation Manager** 控制器后，您必须配置 **KIE** 服务器类加载，以将父类设置为最后加载。

#### 流程

1. 导航到 **Applications** → **Application Types** → **WebSphere Enterprise Applications**，点 **kie-server**。
2. 点左侧 **Details Properties** 标题下的 **Class Loading and Update Detection**。
3. 在属性中，将 **Class Loader Order** 更改为以 **本地类加载的类**（最后一个）和 **WAR Class Loader Policy to Single class loader for application**。
4. 保存对主配置的更改。

#### 49.2. 为无外进程自动化管理器控制器设置系统属性



安装无头进程 Automation Manager 控制器后，在应用程序服务器或服务器中设置本节中列出的系统属性，以启用与无头进程自动化管理器控制器的正确交互。



### 注意

要获得最佳结果，请在生产环境中的不同服务器中安装 KIE 服务器和无头进程自动化管理器控制器。在开发环境中，您可以在同一服务器上安装 KIE 服务器和无头进程自动化管理器控制器。在这两种情况下，请务必在安装了无头进程自动化管理器控制器的所有应用程序服务器上对这些属性进行更改。

### 先决条件

- KIE 服务器和无外处理自动化管理器控制器会在应用服务器实例上安装。

### 流程

1. 在安装了无头进程 Automation Manager 控制器的应用程序服务器实例中指定以下 JVM 属性：

表 49.1. 无外进程自动化管理器控制器所需的属性

Name	要求
<code>org.kie.server.user</code>	具有 <code>kie-server</code> 角色的用户
<code>org.kie.server.pwd</code>	<code>org.kie.server.user</code> 属性中指定的用户的密码

2. 在安装了 KIE Server 的应用程序服务器实例中指定以下 JVM 属性值：

表 49.2. 安装无头进程自动化管理器控制器时，KIE 服务器所需的属性

Name	要求
<code>org.kie.server.controller.user</code>	具有 <code>kie-server</code> 角色的用户
<code>org.kie.server.controller.pwd</code>	为 <code>org.kie.server.controller.user</code> 属性指定的用户密码
<code>org.kie.server.id</code>	KIE 服务器安装的 ID 或名称，如 <code>rhdm700-decision-server-1</code>
<code>org.kie.server.location</code>	KIE 服务器、 <code>http://&lt;HOST&gt;:&lt;PORT&gt;/kie-server/services/rest/server</code> 的 URL

Name	要求
org.kie.server.controller	无头进程自动化管理器控制器的 URL， <b>http://&lt;HOST&gt;:&lt;PORT&gt;/controller/rest/controller</b>

**<HOST >** 是 KIE 服务器主机的 ID 或名称，例如 localhost 或 192.7.8.9。

**<PORT >** 是 KIE 服务器主机的端口号，例如 7001。

### 49.3. 映射无头进程自动化管理器控制器组和角色

安装无头进程 Automation Manager 控制器后，在 WebSphere 集成解决方案控制台中，您必须创建 kie-server 组，然后将 kie-server 角色映射到 kie-server 组。

#### 先决条件

- 一个无头进程自动化管理器控制器安装在 IBM WebSphere Application Server 实例中。

#### 流程

1. 在 WebSphere 集成解决方案控制台中，为该组创建 kie-server 组 and 用户，如创建 **KIE Server 组和角色** 所述。
2. 导航到 **Applications → Application Types → WebSphere Enterprise Applications**，然后选择新安装的无头进程自动化管理器控制器。
3. 在详细属性下，点 **Security Role to User/Group Mapping**。
4. 选择 kie-server 角色并点击 **Map Groups** 搜索 kie-server 组。
5. 将 kie-server 组从 **Available** 列表移到 **Selected** 列表中，然后单击 **OK**。

此映射可让之前创建的管理员用户访问流程自动化管理器。

6.

单击 **Save** 以完成映射。

#### 49.4. 验证安装

安装无头进程自动化管理器控制器并在应用程序服务器上定义所需的系统属性和角色要求，验证无头进程自动化管理器控制器是否正常工作。

##### 先决条件

- **KIE 服务器和无外处理自动化管理器控制器会在应用服务器实例上安装。**
- **您已为应用程序服务器上无头进程 Automation Manager 控制器设置所有必需的系统属性和角色要求。**

##### 流程

在命令终端中，输入以下命令验证无外进程自动化管理器控制器是否正常工作：

```
curl -X GET "http://<HOST>:<PORT>/controller/rest/controller/management/servers" -H "accept: application/xml" -u '<CONTROLLER>:<CONTROLLER_PWD>'
```

**<HOST >** 是无头进程自动化管理器控制器主机的 ID 或名称，如 **localhost** 或 **192.7.8.9**。

**<PORT >** 是无头进程自动化管理器控制器主机的端口号，例如 **7001**。

**<CONTROLLER >** 和 **<CONTROLLER\_PWD >** 是您在此部分中创建的用户凭证。

该命令应该返回有关 **KIE 服务器实例**的信息。



### 注意

另外，您可以使用 **KIE 服务器 Java API 客户端** 访问无头进程自动化管理器控制器。

如果无头进程 **Automation Manager 控制器** 没有运行，请停止和重启应用服务器实例，然后再次尝试访问无头进程自动化管理器控制器 **URL 或 API**。

## 第 50 章 在 IBM WEBSHERE APPLICATION SERVER 中配置嵌入式进程引擎和决策引擎

嵌入式引擎是一个轻量级的工作流程和引擎，可让您执行决策和业务流程。嵌入式引擎可以是 Red Hat Process Automation Manager 应用程序的一部分，或者可以通过 OpenShift、Kubernetes 和 Docker 作为服务来部署。您可以通过 API 或一组上下文和依赖注入(CDI)服务在 Red Hat Process Automation Manager 应用程序中嵌入引擎。

如果要将在嵌入式引擎与 Red Hat Process Automation Manager 应用程序搭配使用，您必须通过将 Red Hat Business Automation 材料(BOM)文件添加到项目的 pom.xml 文件来向项目添加 Maven 依赖项。Red Hat Business Automation BOM 适用于 Red Hat Decision Manager 和 Red Hat Process Automation Manager。有关 Red Hat Automation BOM 的更多信息，请参阅 [Red Hat Process Automation Manager 和 Maven 库版本之间的映射是什么？](#)

## 流程

1. 在 pom.xml 文件中声明 Red Hat Business Automation BOM :

```
<dependencyManagement>
<dependencies>
<dependency>
<groupId>com.redhat.ba</groupId>
<artifactId>ba-platform-bom</artifactId>
<version>7.13.5.redhat-00002</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
<dependencies>
<!-- Your dependencies -->
</dependencies>
```

2. 在 `<dependencies>` 标签中声明项目所需的依赖项。将产品 BOM 导入到项目中后，定义了面向用户的依赖项的版本，因此您无需指定这些 `<dependency>` 元素的 `<version>` 子元素。但是，您必须使用 `<dependency>` 元素来声明您要在项目中使用的依赖项。

- 对于基本的 Red Hat Process Automation Manager 项目，根据您要使用的功能，声明以下依赖项：

嵌入式进程引擎依赖项

```
<!-- Public KIE API -->
```

```
<dependency>
 <groupId>org.kie</groupId>
 <artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-flow</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-flow-builder</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-bpmn2</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-runtime-manager</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-persistence-jpa</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-query-jpa</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-audit</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-kie-services</artifactId>
</dependency>

<!-- Dependency needed for default WorkItemHandler implementations. -->
<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-workitems-core</artifactId>
</dependency>

<!-- Logging dependency. You can use any logging framework compatible with
slf4j. -->
<dependency>
 <groupId>ch.qos.logback</groupId>
```

```

<artifactId>logback-classic</artifactId>
<version>${logback.version}</version>
</dependency>

```

- 对于使用 CDI 的 Red Hat Process Automation Manager 项目，您通常会声明以下依赖项：

#### 启用 CDI 的进程引擎依赖项

```

<dependency>
 <groupId>org.kie</groupId>
 <artifactId>kie-api</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-services-cdi</artifactId>
</dependency>

```

#### 嵌入式决策引擎依赖项

```

<dependency>
 <groupId>org.drools</groupId>
 <artifactId>drools-compiler</artifactId>
</dependency>

<!-- Dependency for persistence support. -->
<dependency>
 <groupId>org.drools</groupId>
 <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>

```

```

<groupId>org.drools</groupId>
<artifactId>drools-decisiontables</artifactId>
</dependency>
<dependency>
<groupId>org.drools</groupId>
<artifactId>drools-templates</artifactId>
</dependency>
<dependency>
<groupId>org.drools</groupId>
<artifactId>drools-scorecards</artifactId>
</dependency>

<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
<groupId>org.kie</groupId>
<artifactId>kie-ci</artifactId>
</dependency>

```

- 要使用 KIE 服务器，声明以下依赖项：

#### 客户端应用程序 KIE 服务器依赖项

```

<dependency>
<groupId>org.kie.server</groupId>
<artifactId>kie-server-client</artifactId>
</dependency>

```

- 要为 Red Hat Process Automation Manager 创建远程客户端，请声明以下依赖项：

#### 客户端依赖项

```

<dependency>
<groupId>org.uberfire</groupId>
<artifactId>uberfire-rest-client</artifactId>
</dependency>

```



- 在创建包含资产的 JAR 文件（如规则和进程定义）时，请将 Maven 项目的打包类型指定为 `kjar`，并使用 `org.kie:kie-maven-plugin` 来处理位于 `< project >` 元素下的 `kjar` 打包类型。在以下示例中，`${kie.version}` 是列出的 Maven 库版本，在 [Red Hat Process Automation Manager 和 Maven 库版本之间有什么映射？](#)

```
<packaging>kjar</packaging>
<build>
 <plugins>
 <plugin>
 <groupId>org.kie</groupId>
 <artifactId>kie-maven-plugin</artifactId>
 <version>${kie.version}</version>
 <extensions>true</extensions>
 </plugin>
 </plugins>
</build>
```

3.

如果您在项目中使用具有持久性支持的进程引擎或决策引擎，您必须通过从 Red Hat Business Automation BOM 文件中复制 `version.org.hibernate-4ee7` 属性，在 `pom.xml` 文件的 `dependencyManagement` 部分中声明以下休眠依赖关系：

#### Hibernate 依赖项

```
<!-- hibernate dependencies -->
<dependencyManagement>
 <dependencies>
 <dependency>
 <groupId>org.hibernate</groupId>
 <artifactId>hibernate-entitymanager</artifactId>
 <version>${version.org.hibernate-4ee7}</version>
 </dependency>

 <dependency>
 <groupId>org.hibernate</groupId>
 <artifactId>hibernate-core</artifactId>
 <version>${version.org.hibernate-4ee7}</version>
 </dependency>
 </dependencies>
</dependencyManagement>
```

## 第 51 章 使用密钥存储保护密码

您可以使用密钥存储来加密用于 **Business Central** 和 **KIE 服务器** 之间的通信的密码。您应该同时对控制器和 **KIE 服务器** 密码进行加密。如果 **Business Central** 和 **KIE 服务器** 部署到不同的应用程序服务器，则两个应用服务器都应该使用密钥存储。

为您的密钥存储使用 **Java Cryptography Extension KeyStore(JCEKS)**，因为它支持对称密钥。



### 注意

如果没有使用 **JCEKS** 配置 **KIE 服务器**，则 **KIE 服务器** 密码以纯文本形式保存在系统属性中。

### 先决条件

- **KIE 服务器** 安装在 **IBM WebSphere Application Server** 中。
- 已创建具有 **kie-server** 角色的 **KIE 服务器** 用户，如 [第 48.1 节“创建 KIE 服务器组和角色”](#) 所述。
- 已安装 **Java 8** 或更高版本。

### 流程

1. 创建 **JCEKS** 密钥存储。
2. 在提示时，输入您创建的 **KIE 服务器** 用户的密码。
3. 设置下表中列出的系统属性：

表 51.1. 用于加载 **KIE 服务器 JCEKS** 的系统属性

系统属性	占位符	描述
------	-----	----

系统属性	占位符	描述
<b>kie.keystore.keyStoreURL</b>	<b>&lt;KEYSTORE_URL&gt;</b>	要使用的 JCEKS 的 URL，如 <b>file:///home/kie/keystores/keystore.jceks</b>
<b>kie.keystore.keyStorePwd</b>	<b>&lt;KEYSTORE_PWD&gt;</b>	JCEKS 的密码
<b>kie.keystore.key.server.alias</b>	<b>&lt;KEY_SERVER_ALIASES&gt;</b>	保存密码的 REST 服务的密钥别名
<b>kie.keystore.key.server.pwd</b>	<b>&lt;KEY_SERVER_PWD&gt;</b>	使用存储的密码的 REST 服务别名的密码
<b>kie.keystore.key.ctrl.alias</b>	<b>&lt;KEY_CONTROL_ALIASES&gt;</b>	存储密码的默认 REST Process Automation Controller 的密钥别名
<b>kie.keystore.key.ctrl.pwd</b>	<b>&lt;KEY_CONTROL_PWD&gt;</b>	默认 REST Process Automation Controller 的别名密码，其存储的密码

4.

启动 KIE 服务器以验证配置。

## 第 52 章 后续步骤

- [决策服务入门](#)
- [使用指导的决策表设计决策服务](#)

## 部分 VI. 在 ORACLE WEBLOGIC 服务器中安装和配置 KIE 服务器

作为系统管理员，您可以为红帽 KIE 服务器配置 Oracle WebLogic Server，并在 Oracle 服务器实例上安装 KIE 服务器。

### 注意

现在，在维护阶段，对 Oracle WebLogic Server 上的 Red Hat Process Automation Manager 的支持。红帽将继续在 Oracle WebLogic 服务器上支持 Red Hat Process Automation Manager，但有以下限制：

- 红帽不会发布新的认证或软件功能。
- 红帽只会发布具有重大影响和关键任务程序错误修复补丁的合格安全修补程序。

未来，红帽可能会将客户定向到与红帽混合云策略兼容的新平台和产品组件。

### 先决条件

- 安装了 Oracle WebLogic Server 实例版本 12.2.1.3.0 或更高版本。有关完整安装说明，请参阅 [Oracle WebLogic Server 产品页面](#)。
- 您可以访问 Oracle WebLogic Server 管理控制台，通常位于 `http://<HOST>:7001/console`。

## 第 53 章 KIE 服务器

**KIE 服务器是保存和运行红帽流程自动化管理器的规则和其他工件的服务器。KIE 服务器是一个独立内置组件，可用于通过可用于 REST、Java 消息服务(JMS)或 Java 客户端平台应用程序的接口实例化和执行规则，以及通过解决者管理进程、作业和红帽构建 OptaPlanner 功能。**

**创建为 Web 可部署 WAR 文件，可以在任何 web 容器中部署 KIE 服务器。当前版本的 KIE 服务器包含在 Red Hat Decision Manager 和 Red Hat Process Automation Manager 的默认扩展中。**

**KIE 服务器占用内存占用量较低，因此可在云中轻松部署。此服务器的每个实例都可以打开并实例化多个容器，可让您并行执行多个规则服务。**

**KIE 服务器可以与其他应用服务器（如 Oracle WebLogic Server 或 IBM WebSphere Application Server）集成，以简化红帽流程自动化管理器应用程序管理。**

## 第 54 章 ORACLE WEBLOGIC SERVER

**Oracle WebLogic Server 是 Java EE 应用服务器，提供标准的 API 集，用于创建分布式 Java 应用程序，可访问各种服务，如数据库、消息服务和与外部企业系统的连接。用户客户端使用 Web 浏览器客户端或 Java 客户端访问这些应用程序。**

## 第 55 章 安装和运行 ORACLE WEBLOGIC SERVER

Oracle WebLogic Server 必须已安装并运行，才能应用适合 KIE 服务器的许多配置。这部分论述了如何在独立 Oracle WebLogic Server 域中安装和启动 Oracle WebLogic Server。

有关最新和详细的安装说明，请参阅 [Oracle WebLogic Server 产品页面](#)。



### 注意

如果您已在运行的 Oracle WebLogic Server 实例，它使用与您要启动的服务器使用的相同侦听器端口，则必须在启动第二服务器前停止第一个服务器。

### 流程

1. 从 Oracle WebLogic Server [下载页面](#) 下载 Oracle WebLogic Server 12.2.1.3.0 或更高版本。
2. 登录目标系统，并验证系统上是否已存在经过认证的 JDK。安装程序需要经过认证的 JDK。有关系统要求，请参阅 [Oracle Fusion 中间件系统要求和规格](#)。要下载 JDK，请参阅 [规划 Oracle Fusion 中间件安装中的 "About JDK requirements" 部分](#)。
3. 进入下载安装程序的目录。
4. 要启动安装程序，从您系统上的 JDK 目录运行 `java -jar`，如下例所示：

在基于 UNIX 的操作系统中，输入以下命令：

```
/home/Oracle/jdk/jdk1.8.0_131/bin/java -jar fmw_12.2.1.3.0_wls_generic.jar
```

在 Windows 操作系统上，输入以下命令：

```
C:\Program Files\Java\jdk1.8.0_131\bin\java -jar fmw_12.2.1.3.0_wls_generic.jar
```

将这些示例中的 JDK 位置替换为您系统上的实际 JDK 位置。



5. 按照安装向导提示完成安装。

6. 安装完成后，导航到 `WLS_HOME/user_projects/<DOMAIN_NAME>` 目录，其中 `&lt;DOMAIN_NAME>` 是域目录。在以下示例中，`mydomain` 是域目录：

```
WLS\user_projects\mydomain
```

7. 输入以下命令之一来启动 Oracle WebLogic Server：

在基于 UNIX 的操作系统中，输入以下命令：

```
startWebLogic.sh
```

在 Windows 操作系统上，输入以下命令：

```
startWebLogic.cmd
```

启动脚本显示一系列信息，最后显示类似于如下的消息：

```
<Dec 8, 2017 3:50:42 PM PDT> <Notice> <WebLogicServer> <000360> <Server started in RUNNING mode>
```

8. 在网页浏览器中打开以下 URL：

```
http://<HOST>:<PORT>/console
```

在这个命令中，替换以下放置拥有者：

- 将 `<HOST >` 替换为主机服务器的系统名称或 IP 地址。
- 将 `&lt;PORT >` 替换为主机服务器侦听请求的端口数量（默认为7001）。

例如，要为系统上运行的 Oracle WebLogic Server 的本地实例启动管理控制台，请在网页浏览器中输入以下 URL：

`http://localhost:7001/console/`

如果使用安全套接字层(SSL)启动管理控制台，您必须在 `http` 之后添加 `s`，如下所示：  
**`https://<HOST>:<PORT>/console`**

9. 出现WebLogic 管理控制台的登录页面时，请输入您的管理凭证。

## 第 56 章 为 KIE 服务器配置 ORACLE WEBLOGIC SERVER

在使用 Oracle WebLogic Server 部署 KIE 服务器之前，您必须在 Oracle WebLogic 服务器上配置系统属性、安全设置、JMS 要求和其他属性。这些配置促进与 KIE 服务器的最佳集成。

### 先决条件

- Oracle WebLogic Server 已安装并运行。
- 已登陆到 WebLogic 管理控制台。

### 56.1. 配置 KIE 服务器组 and 用户

您必须将用户分配给 WebLogic 管理控制台中的 kie-server 组，以便在 Oracle WebLogic Server 中启用容器管理的身份验证机制。

### 流程

1. 在 WebLogic 管理控制台中，单击 **Security Realms**。
2. 选择您所需的安全域，或单击 **New** 以创建新的安全域。
3. 导航到 **Users and Groups** → **Groups** → **New**，再创建 kie-server 组。
4. 导航到 **Users** → **New** 并创建新用户。
5. 输入用户，如 **server-user**，以及此新用户的密码，然后单击确定。



### 重要

确保所选用户名与角色或组的任何已知标题没有冲突。例如，如果有一个名为 **kie-server** 的角色，则不创建一个名为 **kie-server** 的用户。

6. 单击新创建的用户，然后返回到 **Groups** 选项卡。
7. 使用选择工具将 **kie-server** 组从 **Available** 字段移到 **Chosen** 字段，然后单击 **Save**。

## 56.2. 在 ORACLE WEBLOGIC SERVER 中配置 JDBC 数据源

数据源是一个对象，它允许 Java 数据库连接(JDBC)客户端（如应用服务器）与数据库建立连接。应用程序在 Java 命名和目录接口(JNDI)树或本地应用程序上下文中查找数据源，并请求数据库连接来检索数据。您必须为 Oracle WebLogic Server 配置数据源，以确保在服务器和指定数据库之间进行适当的数据交换。

通常，使用 Red Hat Process Automation Manager 的解决方案在单个事务中管理多个资源。用于异步作业、事件和计时器的 JMS，例如：在确保数据原子性和一致性结果时，Red Hat Process Automation Manager 需要数据源中的 XA 驱动程序。如果监听器内存在针对不同架构的事务代码，或者从 jBPM 引擎提供的 hook 派生出来，则还需要一个 XA 驱动程序。

不要使用非 XA 数据源，除非您正面没有参与单一事务的多个资源。

### 先决条件

- 您要用于创建数据库连接的 JDBC 驱动程序会在您要部署数据源的所有服务器上安装。一些 JDBC 驱动程序安装在 Oracle WebLogic Server 中，如用于 DB2 的 WebLogic 品牌数据直接 JDBC 驱动程序、Informix、MS SQL Server 和 Sybase。有关 JDBC 驱动程序的更多信息，请参阅 Oracle 帮助中心中的[将 JDBC 驱动程序与 WebLogic 服务器搭配使用](#)。

### 流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
  - 产品：流程自动化管理器
  - Version: 7.13.5
2. 下载 Red Hat Process Automation Manager 7.13.5 附加组件。

3.

完成以下步骤以准备数据库：

a.

在临时目录中提取 `rhpm-7.13.5-add-ons.zip`，如 `TEMP_DIR`。

b.

提取 `TEMP_DIR/rhpm-7.13.5-migration-tool.zip`。

c.

将您的当前目录更改为 `TEMP_DIR/rhpm-7.13.5-migration-tool/ddl-scripts` 目录。此目录包含多种数据库类型的 DDL 脚本。

d.

将您的数据库类型的 DDL 脚本导入到您要使用的数据库中，例如：

```
psql jbpn < /ddl-scripts/postgresql/postgresql-jbpn-schema.sql
```



注意

如果您与 Spring Boot 结合使用 PostgreSQL 或 Oracle，则必须导入对应的 Spring Boot DDL 脚本，如 `/ddl-scripts/oracle/oracle-springboot-jbpn-schema.sql` 或 `/ddl-scripts/postgresql/postgresql-springboot-jbpn-schema.sql`。

4.

在 WebLogic 管理控制台中，导航到 `Change Center` → `Lock & Edit`。

5.

在 `Domain Structure` 下，点 `Services` → `Data Sources`。

6.

在 `Summary of Data Sources` 页面中，点击 `New` → `Generic Data Source`。

7.

在 `JDBC Data Sources Properties` 页面中，输入或选择以下信息：

•

**Name**：输入此 JDBC 数据源的名称。每当引用此数据源时，此名称在配置文件 (`config.xml`) 中以及整个管理控制台中使用。

•

**JNDI Name**：输入要绑定到此 JDBC 数据源的 JNDI 路径。应用程序在保留连接时，按此名称查找 JNDI 树上的数据源。

- **数据库类型**：选择您要连接的数据库的 **DBMS**。如果没有列出您的 **DBMS**，请选择"其他"。
8. 点 **Next** 继续。
9. 选择您要用来连接到数据库的数据库驱动程序。该列表包含所选 **DBMS** 的通用 **JDBC** 驱动程序，以及之前安装的任何其他 **JDBC** 驱动程序。
10. 在 **Transaction Options** 页面中，保留选中 **Supports Global Transactions** 选项，然后从可用的事务选项中选择。您还可以清除此复选框以在这个数据源中禁用（忽略）全局事务。在大多数情况下，您应保留选择的选项以实现最佳数据效率。
- **two-Phase Commit**：选择该选项以启用标准 **XA** 处理。只有在选择 **XA JDBC** 驱动程序进行数据库连接时，此选项才可用。
  - **Logging Last Resource**：选择这个选项，以启用非 **XA JDBC** 连接，以使用日志记录 **Last Resource(LLR)**事务优化进行全局事务。建议使用这个选项替换两个 **Phase Commit**。只有在选择非 **XA JDBC** 驱动程序进行数据库连接时，此选项才可用。
  - **模拟双Phase Commit**：选择此选项以启用非 **XA JDBC** 连接，以模拟使用 **JTA** 进行分布式事务的参与。只有在应用程序可以容忍 **Heuristic** 条件时，才选择此选项。只有在选择非 **XA JDBC** 驱动程序进行数据库连接时，此选项才可用。
  - **One-Phase Commit**：选择此选项可启用非 **XA** 连接，以参与到全球交易中，作为唯一的交易参与者。只有在选择非 **XA JDBC** 驱动程序进行数据库连接时，此选项才可用。
11. 点 **Next** 继续。
12. 在 **Connection Properties** 页面中，输入以下属性的值：
- **service Name**：指定要连接到的数据库的服务名称。如果提供多个数据源，则每个数据源必须相同。只有在为 **Oracle Real Application Clusters(RAC)**选择了可用 **service-instance** 连接驱动程序之一时，此字段才可用。

- **数据库名称**：输入您要连接的数据库名称。确切的数据库名称要求因 JDBC 驱动程序和 DBMS 而异。
- **主机名**：输入托管数据库的服务器的 DNS 名称或 IP 地址。如果您要创建 Oracle GridLink service-instance 连接，如果提供多个数据源，则每个数据源必须相同。
- **端口**：输入数据库服务器侦听连接请求的端口。
- **数据库用户名**：输入您要用于数据源中的每个连接的数据库用户帐户名称。
- **password /Confirm 密码**：输入数据库用户帐户的密码。
- **Oracle.jdbc.DRCPConnectionClass**：可选，输入您的环境需要的数据库 Resident Connection Pooling(DCRP)连接类。

13.

点 **Next** 继续。

14.

在 **Test Database Connection** 页面上，查看连接参数，再单击 **Test Configuration**。

**Oracle WebLogic Server** 尝试创建从管理服务器到数据库的连接。连接测试的结果显示在页面的顶部。如果测试失败，请更正所有配置错误并重试测试。

15.

如果没有在管理服务器上安装 JDBC 驱动程序，则单击 **Next** 以继续或跳过这一步。

16.

在 **Select Targets** 页面中，选择要在其上部署数据源的服务器或集群，然后单击 **Finish**。

17.

返回 WebLogic 管理控制台的主菜单并选择 **Change Center** → **Activate Changes**。

有关 Oracle WebLogic 服务器数据源的更多信息，请参阅 Oracle Help Center 中的 [Oracle WebLogic Server 的 JDBC 数据源](#)。

## 56.3. 配置 JAVA 消息服务(JMS)

Java 消息服务(JMS)是一个 Java API, KIE 服务器用来与其他应用服务器 (如 Oracle WebLogic Server 和 IBM WebSphere Application Server) 交换消息。您必须将应用服务器配置为通过 KIE 服务器发送和接收 JMS 消息, 以确保两个服务器之间的协作。

### 56.3.1. 创建 JMS 服务器

创建 JMS 服务器以使用 JMS 与 KIE 服务器和 Oracle WebLogic Server。

#### 流程

1. 在 WebLogic 管理控制台中, 导航到 **Services** → **Messaging** → **JMS Servers**。
2. 单击 **New** 以创建新的 JMS 服务器。
3. 为您的 JMS 服务器输入一个名称, 然后单击下一步。
4. 选择为 KIE 服务器部署选择的目标服务器。
5. 点 **Finish**。

### 56.3.2. 创建 JMS 模块

您必须创建一个 JMS 模块来存储 JMS 资源, 如连接工厂和队列。

#### 先决条件

- 您已创建了 JMS 服务器。

#### 流程

1. 在 WebLogic 管理控制台中, 导航到 **Services** → **Messaging** → **JMS Modules**。



2. 单击**新建** 以创建模块。
3. 输入模块名称并点 **Next**。
4. 选择为 **KIE 服务器部署**选择的目标服务器，然后单击"完成"。
5. 单击新创建的模块名称，然后单击 **Subdeployments**。
6. 单击 **New**，为您的模块创建子部署。
7. 为您的子部署指定一个名称，再单击 **Next**。
8. 选择 选择之前创建的 **JMS 服务器**的复选框。
9. 单击 **Finish** 以完成子部署配置。

### 56.3.3. 创建 JMS 连接工厂

要使用 KIE 服务器启用消息，您必须创建某些 **JMS 连接工厂** 以发送和接收消息。

#### 先决条件

- 您已创建了 **JMS 服务器**。
- 您已创建了 **JMS 模块**。

#### 流程

1. 在 **WebLogic 管理控制台**中，导航到 **Services** → **Messaging** → **JMS Modules** 以查看 **JMS 模块**的列表。

2. 选择您之前创建的模块，再单击 **New** 以创建新的 **JMS** 资源。
3. 选择 **Connection Factory**，然后单击下一步。
4. 对于下表中列出的每一个因素，请输入连接工厂的名称（例如：**KIE.SERVER.REQUEST**）和 **JNDI** 名称（如 **jms/cf/KIE.SERVER.REQUEST**）并单击 **Next**。连接工厂自动选择分配给 **JMS** 模块的服务器作为默认值。

表 56.1. KIE 服务器所需的 JMS 连接工厂

Name	默认值	用于
<b>KIE.SERVER.REQUEST</b>	<b>jms/cf/KIE.SERVER.REQUEST</b>	将所有请求发送到 KIE 服务器
<b>KIE.SERVER.RESPONSE</b>	<b>jms/cf/KIE.SERVER.RESPONSE</b>	接收 KIE 服务器生成的所有响应
<b>KIE.SERVER.EXECUTOR</b>	<b>jms/cf/KIE.SERVER.EXECUTOR</b>	KIE 服务器执行程序服务

5. 点 **Finish** 添加连接工厂，并重复每个所需工厂。

#### 56.3.4. 创建 JMS 队列

**JMS** 队列是点对点消息传递的目标端点。您必须创建特定的 **JMS** 队列，以启用使用 **KIE** 服务器的 **JMS** 消息传递。

##### 先决条件

- 您已创建了 **JMS** 服务器。
- 您已创建了 **JMS** 模块。

##### 流程

1. 在 **WebLogic** 管理控制台中，导航到 **Services** → **Messaging** → **JMS Modules** 以查看 **JMS** 模块的列表。

2. 选择您之前创建的模块，然后单击**新建**以创建新的 **JMS** 资源。
3. 选择 **Queue**，再单击 **Next**。
4. 对于下表中列出的每个所需队列，请输入队列的名称（如 **KIE.SERVER.REQUEST**）和 **JNDI** 名称（如 **jms/KIE.SERVER.REQUEST**），然后单击 **Next**。

表 56.2. KIE 服务器所需的 JMS 队列

Name	默认值	用于
<b>KIE.SERVER.REQUEST</b>	<b>jms/KIE.SERVER.REQUEST</b>	将所有请求发送到 KIE 服务器
<b>KIE.SERVER.RESPONSE</b>	<b>JMS/KIE.SERVER.RESPONSE</b>	接收 KIE 服务器生成的所有响应
<b>KIE.SERVER.EXECUTOR</b>	<b>JMS/KIE.SERVER.EXECUTOR</b>	KIE 服务器执行程序服务

5. 选择连接到 **JMS** 服务器的 **JMS** 模块子部署。
6. 点 **Finish** 添加队列，然后对每个所需队列重复。

#### 56.4. 在 ORACLE WEBLOGIC SERVER 中设置系统属性

在部署 KIE 服务器前，设置本部分中列出的系统属性。

##### 流程

1. 设置以下系统属性以增加 **Java** 虚拟机(JVM)内存大小：

```
USER_MEM_ARGS=-Xms512m -Xmx1024m
```

如果您没有增加 **JVM** 内存大小，**Oracle WebLogic Server freezes** 或在部署 **KIE** 服务器时导致部署错误。

2.

在 Oracle WebLogic 服务器实例上指定 KIE 服务器的以下系统属性：

表 56.3. KIE 服务器的系统属性

Name	值	描述
<b>kie.server.jms.queues.response</b>	<b>JMS/KIE.SERVER.RESPONSE</b>	用于 KIE 服务器使用的响应的 JMS 队列的 JNDI 名称。
<b>org.kie.server.domain</b>	<b>OracleDefaultLoginConfiguration</b>	在使用 JMS 时用于验证用户的身份的 JAAS <b>LoginContext</b> 域。
<b>org.kie.server.persistence.ds</b>	<b>jdbc/jbpm</b>	KIE 服务器的数据源 JNDI 名称。
<b>org.kie.server.persistence.tm</b>	<b>org.hibernate.service.jta.platform.internal.WeblogicJtaPlatform</b>	用于设置 Hibernate 属性的事务管理器平台。
<b>org.kie.server.persistence.dialect</b>	示例： <b>org.hibernate.dialect.H2Dialect</b>	指定要使用的 Hibernate 权利。根据数据源设置。
<b>org.kie.executor.jms.queue</b>	<b>JMS/KIE.SERVER.EXECUTOR</b>	KIE 服务器的作业执行器 JMS 队列。
<b>org.kie.executor.jms.cf</b>	<b>jms/cf/KIE.SERVER.EXECUTOR</b>	KIE 服务器的作业执行器 JMS 连接工厂。
<b>org.kie.server.router</b>	示例： <b>http://localhost:9000</b>	(可选) 指定应用服务器在集群 KIE 服务器环境中包含一个或多个 KIE 服务器路由器(Smart Router)的一个或多个 URL。

3.

在 **JAVA\_OPTIONS** 环境变量中设置相同的属性值：

```

JAVA_OPTIONS="-Dkie.server.jms.queues.response=jms/KIE.SERVER.RESPONSE
-Dorg.kie.server.domain=OracleDefaultLoginConfiguration
-Dorg.kie.executor.jms.cf=jms/cf/KIE.SERVER.EXECUTOR
-Dorg.kie.executor.jms.queue=jms/KIE.SERVER.EXECUTOR
-Dorg.kie.server.persistence.ds=jdbc/jbpm
-
Dorg.kie.server.persistence.tm=org.hibernate.service.jta.platform.internal.WeblogicJtaPlatform
-Dorg.kie.server.persistence.dialect=org.hibernate.dialect.H2Dialect
// Optional server router, for clustered server environment
-Dorg.kie.server.router=http://localhost:9000

```

## 56.5. 停止和重启 ORACLE WEBLOGIC SERVER

在 Oracle WebLogic Server 中配置了所有必需的系统属性后，停止并重启 Oracle 服务器，以确保应用了配置。

### 流程

1. 在 WebLogic 管理控制台中，导航到 **Change Center** → **Lock & Edit**.
2. 在 **Domain Structure** 下，点击 **Environment** → **Servers** → **Control**.
3. 选择您要停止的服务器，然后单击 **关闭**。
4. 选择 **Work Completes** 以正常关闭服务器，或选择 **Force Shutdown Now** 来立即停止服务器，而无需完成持续的任务。
5. 在 **Server Life Cycle Assistant** 窗格中，点 **Yes** 以完成关闭。
6. 关闭完成后，导航到命令终端中的域目录 `WLS_HOME/user_projects/<DOMAIN_NAME>`。例如：  

```
WLS\user_projects\mydomain
```
7. 输入以下命令重启 Oracle WebLogic Server 以应用新配置之一：  
  
在基于 **UNIX** 的操作系统中：  

```
startWebLogic.sh
```

  
在 **Windows** 操作系统中：  

```
startWebLogic.cmd
```
8. 在网页浏览器中打开管理控制台（例如 `http://localhost:7001/console/`），并使用您的凭证

登录。

## 第 57 章 使用 ORACLE WEBLOGIC SERVER 安装 KIE 服务器

在 Oracle WebLogic Server 中配置了所有必需的系统属性后，您可以使用 Oracle WebLogic Server 安装 KIE 服务器，以简化红帽流程自动化管理器应用程序管理。

### 先决条件

- 甲骨文 WebLogic 服务器实例进行了配置，如 [第 56 章 为 KIE 服务器配置 Oracle WebLogic Server](#) 所述。

### 流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
  - 产品：流程自动化管理器
  - Version: 7.13.5
2. 下载 适用于所有支持的 EE7 容器的 Red Hat Process Automation Manager 7.13.5 KIE Server。
3. 将下载的 `rhpm-7.13.5-kie-server-ee7.zip` 文件提取到临时目录中。
4. 在 WebLogic 管理控制台中，导航到 **Deployments** 以查看所有现有应用程序。
5. 点 **Install**。
6. 导航到您下载并提取 `rhpm-7.13.5-kie-server-ee7.zip` 文件的临时目录，再前往 `rhpm-7.13.5-kie-server-ee7/kie-server.war`。
7. 选择 `kie-server.war` 文件并单击“下一步”以继续。

8. 选择将这个部署作为目标风格的应用程序，然后点击 **Next**。
9. 将应用程序名称设置为 **kie-server**，并将安全模型设置为 **DD**（仅限 **DD**）。其余选项保留默认值，然后单击“下一步”以继续。
10. 在 **Additional Configuration** 部分中，选择 **No**，然后在稍后检查配置 并点击 **Finish**。

### 57.1. 在 ORACLE WEBLOGIC SERVER 上验证 KIE 服务器安装

在 Oracle WebLogic Server 上安装 KIE 服务器后，验证安装是否成功。

#### 先决条件

- 甲骨文 WebLogic 服务器实例进行了配置，如 [第 56 章 为 KIE 服务器配置 Oracle WebLogic Server](#) 所述。
- 如 [第 57 章 使用 Oracle WebLogic Server 安装 KIE 服务器](#) 所述安装 KIE 服务器。

#### 流程

1. 在网页浏览器中输入 KIE 服务器 URL `http://<HOST>:<PORT>/kie-server/services/rest/server`。
2. 验证 KIE 服务器是否正在运行。

如果 KIE 服务器没有运行，请停止和重启 Oracle WebLogic Server 实例，并尝试再次尝试访问 KIE 服务器 URL。



## 第 58 章 使用 ORACLE WEBLOGIC SERVER 安装并运行无头进程自动化管理器控制器

要使用 KIE 服务器 REST API 或 Java Client API 与 KIE 服务器交互，请使用 Oracle WebLogic Server 安装无头进程自动化管理器控制器。无外进程自动化管理器控制器以集中的方式管理 KIE 服务器配置，以便您可以使用无外进程自动化管理器控制器创建和维护容器并执行其他服务器级任务。

### 先决条件

- 甲骨文 WebLogic 服务器实例配置如 [第 56 章 为 KIE 服务器配置 Oracle WebLogic Server](#) 所述。
- KIE 服务器安装在 Oracle WebLogic 服务器实例上。
- 有足够的用户权限才能完成安装。

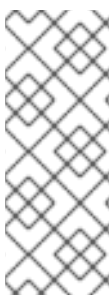
### 流程

1. 进入红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本：
  - 产品：流程自动化管理器
  - Version: 7.13.5
2. 下载 Red Hat Process Automation Manager 7.13.5 附加组件。
3. 将下载的 `rhpm-7.13.5-add-ons.zip` 文件提取到临时目录中。
4. 在 WebLogic 管理控制台中，导航到 **Security Realms** → **Users and Groups**。
5. 在之前创建的 `kie-server` 组中，为无头进程自动化管理器控制器创建一个用户，如 `控制器`，以及这个新用户的密码，然后单击确定。有关创建组和用户的详情，请参考 [第 56.1 节 “配置 KIE 服务器组 and 用户”](#)。

6. 前往 **Deployments** 以查看所有现有应用程序。
7. 点 **Install**。
8. 导航到您下载并提取 `rhcam-7.13.5-add-ons.zip` 文件的临时目录，再前往 `rhcam-7.13.5-add-ons/rhcam-7.13.5-controller-ee7.zip/controller.war`。
9. 选择 `controller.war` 文件，再单击 **Next** 以继续。
10. 选择将这个部署作为目标风格的应用程序，然后单击 **Next**。
11. 将应用程序名称保留为 **控制器**，仅将安全模型设置为 **DD**。其余选项保留默认值，然后单击"下一步"以继续。
12. 在 **Additional Configuration** 部分中，选择 **No**，然后在稍后检查配置 并单击 **Finish**。

### 58.1. 为无外进程自动化管理器控制器设置系统属性

安装无头进程 **Automation Manager** 控制器后，在应用程序服务器或服务器中设置本节中列出的系统属性，以启用与无头进程自动化管理器控制器的正确交互。



#### 注意

要获得最佳结果，请在生产环境中的不同服务器中安装 **KIE** 服务器和无头进程自动化管理器控制器。在开发环境中，您可以在同一服务器上安装 **KIE** 服务器和无头进程自动化管理器控制器。在这两种情况下，请务必在安装了无头进程自动化管理器控制器的所有应用程序服务器上对这些属性进行更改。

#### 先决条件

- **KIE** 服务器和无外处理自动化管理器控制器会在应用服务器实例上安装。

#### 流程

- 1.

在安装了无头进程 Automation Manager 控制器的应用程序服务器实例中指定以下 JVM 属性值：

表 58.1. 无外进程自动化管理器控制器所需的属性

Name	要求
<b>org.kie.server.user</b>	具有 <b>kie-server</b> 角色的用户
<b>org.kie.server.pwd</b>	<b>org.kie.server.user</b> 属性中指定的用户的密码

2.

在安装了 KIE Server 的应用程序服务器实例中指定以下 JVM 属性值：

表 58.2. 安装无头进程自动化管理器控制器时，KIE 服务器所需的属性

Name	要求
<b>org.kie.server.controller.user</b>	具有 <b>kie-server</b> 角色的用户
<b>org.kie.server.controller.pwd</b>	为 <b>org.kie.server.controller.user</b> 属性指定的用户密码
<b>org.kie.server.id</b>	KIE 服务器安装的 ID 或名称，如 <b>rhdm700-decision-server-1</b>
<b>org.kie.server.location</b>	KIE 服务器、 <b>http://&lt;HOST&gt;:&lt;PORT&gt;/kie-server/services/rest/server</b> 的 URL
<b>org.kie.server.controller</b>	无头进程自动化管理器控制器的 URL， <b>http://&lt;HOST&gt;:&lt;PORT&gt;/controller/rest/controller</b>

<HOST > 是 KIE 服务器主机的 ID 或名称，例如 localhost 或 192.7.8.9。

<PORT > 是 KIE 服务器主机的端口号，例如 7001。

## 58.2. 验证安装

安装无头进程自动化管理器控制器并在应用程序服务器上定义所需的系统属性和角色要求，验证无头进程自动化管理器控制器是否正常工作。

### 先决条件

- **KIE 服务器和无外处理自动化管理器控制器会在应用服务器实例上安装。**
- **您已为应用程序服务器上无头进程 Automation Manager 控制器设置所有必需的系统属性和角色要求。**

## 流程

在命令终端中，输入以下命令验证无外进程自动化管理器控制器是否正常工作：

```
curl -X GET "http://<HOST>:<PORT>/controller/rest/controller/management/servers" -H "accept: application/xml" -u '<CONTROLLER>:<CONTROLLER_PWD>'
```

**<HOST >** 是无头进程自动化管理器控制器主机的 ID 或名称，如 `localhost` 或 `192.7.8.9`。

**<PORT >** 是无头进程自动化管理器控制器主机的端口号，例如 `7001`。

**<CONTROLLER >** 和 **<CONTROLLER\_PWD >** 是您在此部分中创建的用户凭证。

该命令应该返回有关 KIE 服务器实例的信息。



### 注意

另外，您可以使用 KIE 服务器 Java API 客户端访问无头进程自动化管理器控制器。

如果无头进程 Automation Manager 控制器没有运行，请停止和重启应用服务器实例，然后再次尝试访问无头进程自动化管理器控制器 URL 或 API。

## 第 59 章 在 ORACLE WEBLOGIC SERVER 中配置嵌入式进程引擎和决策引擎

嵌入式引擎是一个轻量级的工作流程和引擎，可让您执行决策和业务流程。嵌入式引擎可以是 Red Hat Process Automation Manager 应用程序的一部分，或者可以通过 OpenShift、Kubernetes 和 Docker 作为服务来部署。您可以通过 API 或一组上下文和依赖注入(CDI)服务在 Red Hat Process Automation Manager 应用程序中嵌入引擎。

如果要将在嵌入式引擎与 Red Hat Process Automation Manager 应用程序搭配使用，您必须通过将 Red Hat Business Automation 材料(BOM)文件添加到项目的 pom.xml 文件来向项目添加 Maven 依赖项。Red Hat Business Automation BOM 适用于 Red Hat Decision Manager 和 Red Hat Process Automation Manager。有关 Red Hat Automation BOM 的更多信息，请参阅 [Red Hat Process Automation Manager 和 Maven 库版本之间的映射是什么？](#)

## 流程

1. 在 pom.xml 文件中声明 Red Hat Business Automation BOM :

```
<dependencyManagement>
<dependencies>
<dependency>
<groupId>com.redhat.ba</groupId>
<artifactId>ba-platform-bom</artifactId>
<version>7.13.5.redhat-00002</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
<dependencies>
<!-- Your dependencies -->
</dependencies>
```

2. 在 `<dependencies>` 标签中声明项目所需的依赖项。将产品 BOM 导入到项目中后，定义了面向用户的依赖项的版本，因此您无需指定这些 `<dependency>` 元素的 `<version>` 子元素。但是，您必须使用 `<dependency>` 元素来声明您要在项目中使用的依赖项。

- 对于基本的 Red Hat Process Automation Manager 项目，根据您要使用的功能，声明以下依赖项：

嵌入式进程引擎依赖项

```
<!-- Public KIE API -->
```

```
<dependency>
 <groupId>org.kie</groupId>
 <artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-flow</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-flow-builder</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-bpmn2</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-runtime-manager</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-persistence-jpa</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-query-jpa</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-audit</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-kie-services</artifactId>
</dependency>

<!-- Dependency needed for default WorkItemHandler implementations. -->
<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-workitems-core</artifactId>
</dependency>

<!-- Logging dependency. You can use any logging framework compatible with
slf4j. -->
<dependency>
 <groupId>ch.qos.logback</groupId>
```

```
<artifactId>logback-classic</artifactId>
<version>${logback.version}</version>
</dependency>
```

- 对于使用 CDI 的 Red Hat Process Automation Manager 项目，您通常会声明以下依赖项：

#### 启用 CDI 的进程引擎依赖项

```
<dependency>
 <groupId>org.kie</groupId>
 <artifactId>kie-api</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
 <artifactId>jbpm-services-cdi</artifactId>
</dependency>
```

#### 嵌入式决策引擎依赖项

```
<dependency>
 <groupId>org.drools</groupId>
 <artifactId>drools-compiler</artifactId>
</dependency>

<!-- Dependency for persistence support. -->
<dependency>
 <groupId>org.drools</groupId>
 <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>
```

```

<groupId>org.drools</groupId>
<artifactId>drools-decisiontables</artifactId>
</dependency>
<dependency>
<groupId>org.drools</groupId>
<artifactId>drools-templates</artifactId>
</dependency>
<dependency>
<groupId>org.drools</groupId>
<artifactId>drools-scorecards</artifactId>
</dependency>

<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
<groupId>org.kie</groupId>
<artifactId>kie-ci</artifactId>
</dependency>

```

- 要使用 KIE 服务器，声明以下依赖项：

#### 客户端应用程序 KIE 服务器依赖项

```

<dependency>
<groupId>org.kie.server</groupId>
<artifactId>kie-server-client</artifactId>
</dependency>

```

- 要为 Red Hat Process Automation Manager 创建远程客户端，请声明以下依赖项：

#### 客户端依赖项

```

<dependency>
<groupId>org.uberfire</groupId>
<artifactId>uberfire-rest-client</artifactId>
</dependency>

```



- 在创建包含资产的 JAR 文件（如规则和进程定义）时，请将 Maven 项目的打包类型指定为 `kjar`，并使用 `org.kie:kie-maven-plugin` 来处理位于 `<project>` 元素下的 `kjar` 打包类型。在以下示例中，`${kie.version}` 是列出的 Maven 库版本，在 [Red Hat Process Automation Manager 和 Maven 库版本之间有什么映射？](#)

```
<packaging>kjar</packaging>
<build>
 <plugins>
 <plugin>
 <groupId>org.kie</groupId>
 <artifactId>kie-maven-plugin</artifactId>
 <version>${kie.version}</version>
 <extensions>true</extensions>
 </plugin>
 </plugins>
</build>
```

- 如果您在项目中具有持久性支持的进程引擎或决策引擎，您必须通过从 Red Hat Business Automation BOM 文件中复制 `version.org.hibernate-4ee7` 属性，在 `pom.xml` 文件的 `dependencyManagement` 部分中声明以下休眠依赖关系：

#### Hibernate 依赖项

```
<!-- hibernate dependencies -->
<dependencyManagement>
 <dependencies>
 <dependency>
 <groupId>org.hibernate</groupId>
 <artifactId>hibernate-entitymanager</artifactId>
 <version>${version.org.hibernate-4ee7}</version>
 </dependency>

 <dependency>
 <groupId>org.hibernate</groupId>
 <artifactId>hibernate-core</artifactId>
 <version>${version.org.hibernate-4ee7}</version>
 </dependency>
 </dependencies>
</dependencyManagement>
```

## 第 60 章 使用密钥存储保护密码

您可以使用密钥存储来加密用于 Business Central 和 KIE 服务器之间的通信的密码。您应该同时对控制器和 KIE 服务器密码进行加密。如果 Business Central 和 KIE 服务器部署到不同的应用程序服务器，则两个应用服务器都应该使用密钥存储。

为您的密钥存储使用 Java Cryptography Extension KeyStore(JCEKS)，因为它支持对称密钥。使用 KeyTool 作为 JDK 安装的一部分，可以创建一个新的 JCEKS。



### 注意

如果没有使用 JCEKS 配置 KIE 服务器，则 KIE 服务器密码以纯文本形式保存在系统属性中。

### 先决条件

- KIE 服务器安装在 Oracle WebLogic Server 中。
- 已创建具有 kie-server 角色的 KIE 服务器用户，如 [第 56.1 节“配置 KIE 服务器组 and 用户”](#) 所述。
- 已安装 Java 8 或更高版本。

### 流程

1. 要使用 KeyTool 创建 JCEKS，请在 Java 8 主目录中输入以下命令：

```
$<JAVA_HOME>/bin/keytool -importpassword -keystore <KEYSTORE_PATH> -
keypass <ALIAS_KEY_PASSWORD> -alias <PASSWORD_ALIAS> -storepass
<KEYSTORE_PASSWORD> -storetype JCEKS
```

在本例中，替换以下变量：

- `<KEYSTORE_PATH >` : 存储密钥存储的路径

- <KEYSTORE\_PASSWORD> : 密钥存储密码
  - <ALIAS\_KEY\_PASSWORD > : 用于访问使用别名存储的值的密码
  - <PASSWORD\_ALIAS > : 条目的别名
2. 在提示时, 输入您创建的 KIE 服务器用户的密码。
  3. 设置下表中列出的系统属性 :

表 60.1. 用于加载 KIE 服务器 JCEKS 的系统属性

系统属性	占位符	描述
<b>kie.keystore.keyStoreURL</b>	<KEYSTORE_URL>	要使用的 JCEKS 的 URL, 如 <b>file:///home/kie/keystores/keystore.jceks</b>
<b>kie.keystore.keyStorePwd</b>	<KEYSTORE_PWD>	JCEKS 的密码
<b>kie.keystore.key.server.alias</b>	<KEY_SERVER_ALIASES>	保存密码的 REST 服务的密钥别名
<b>kie.keystore.key.server.pwd</b>	<KEY_SERVER_PWD>	使用存储的密码的 REST 服务别名的密码
<b>kie.keystore.key.ctrl.alias</b>	<KEY_CONTROL_ALIASES>	存储密码的默认 REST Process Automation Controller 的密钥别名
<b>kie.keystore.key.ctrl.pwd</b>	<KEY_CONTROL_PWD>	默认 REST Process Automation Controller 的别名密码, 其存储的密码

4. 启动 KIE 服务器以验证配置。

## 第 61 章 后续步骤

- [决策服务入门](#)
- [使用指导的决策表设计决策服务](#)

## 部分 VII. 修补和升级红帽流程自动化管理器

作为 Red Hat Process Automation Manager 用户，您可以在 Red Hat Process Automation Manager 中应用更新，因为它们在红帽客户门户网站中可用，以使用最新的功能增强和修补程序保持 Red Hat Process Automation Manager 版本。红帽为新产品发行版本提供了更新工具和产品通知，因此您可以更轻松地对 Red Hat Process Automation Manager 和 KIE 服务器应用有用的更新。

### 先决条件

- 您有红帽客户门户网站帐户。
- 安装了 Red Hat Process Automation Manager。有关安装选项，请参阅 [规划 Red Hat Process Automation Manager 安装](#)。



### 注意

如果您使用 Red Hat Process Automation Manager 7.10，而您想要升级到 Red Hat Process Automation Manager 7.13，请参阅 [补丁和升级 Red Hat Process Automation Manager 7.10](#)。

## 第 62 章 RED HAT PROCESS AUTOMATION MANAGER 版本

Red Hat Process Automation Manager 版本使用数字 Major.Minor.Patch 格式指定，如 7.5.1。在本例中，主发行版本为 7.x.x，次版本为 7.5.x，补丁发行版本为 7.5.1。主发行版本通常需要数据迁移，而次版本升级和补丁更新通常通过 Red Hat Process Automation Manager 发行工件提供的更新工具进行管理。

以下是 Red Hat Process Automation Manager 的一般发行版本类型：

### 主发行版本迁移

Red Hat Process Automation Manager 的主要发行版本包括大量改进、安全更新、错误修复以及可能重新设计的功能和功能。当应用程序从一个主版本移动到另一个主版本（如从 Red Hat JBoss BPM Suite 6.4.x）迁移到 Red Hat Process Automation Manager 7.0 时，通常需要数据迁移。自动化迁移工具通常附带新的 Red Hat Process Automation Manager 主版本，以便促进迁移，但某些数据和配置可能需要一些手动工作。支持的迁移路径在产品公告和文档中指定。

有关迁移说明，[请参阅从 Red Hat JBoss BPM Suite 6.4 迁移到 Red Hat Process Automation Manager 7.0。](#)

### 次发行版本升级

Red Hat Process Automation Manager 的次发行版本包括增强、安全更新和程序错误修复。当应用程序从一个次版本移到另一个次版本时，可能需要数据迁移，如从 Red Hat Process Automation Manager 7.5.x 升级到 7.6。Red Hat Process Automation Manager 的补丁更新和新的次要版本中通常会提供自动更新，以帮助更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE 服务器和无头进程自动化管理器控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）都会作为新的工件发布，且您必须重新安装它们以应用更新。

升级到新的次版本之前，请将最新的补丁更新应用到 Red Hat Process Automation Manager 的最新版本，以确保次要发行本升级成功。

### 补丁更新

Red Hat Process Automation Manager 的补丁更新包括最新的安全更新和程序错误修复。计划的补丁更新包含之前为该产品次要版本发布的所有补丁更新，因此您无需逐步应用各个补丁更新，以应用最新的更新。例如，您可以将 Red Hat Process Automation Manager 7.5.0 或 7.5.1 更新至 Red Hat Process Automation Manager 7.5.2。但是，为了获得最佳 Red Hat Process Automation Manager 性能，在产品可用时应用产品更新。有时，红帽可能会在现有产品的正常更新周期外发布非计划补丁更新。这可能包括安全性或其他由 Red Hat 全球支持服务(GSS)提供的更新，以便修复特定问题，且可能不是累积更新。

Red Hat Process Automation Manager 的补丁更新和新的次要版本中通常会提供自动更新，以帮助更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE 服务器

---

和无头进程自动化管理器控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）都会作为新的工件发布，且您必须重新安装它们以应用更新。

为确保在版本间进行最佳过渡，并使用最新的改进和修补程序保持 Red Hat Process Automation Manager，在红帽客户门户网站中可用，对 Red Hat Process Automation Manager 应用新的产品版本和更新。请考虑在红帽客户门户网站中启用产品通知。

## 第 63 章 在红帽客户门户网站中启用通知

您可以在红帽客户门户网站中启用通知以接收产品更新和公告。这些通知会通知您更新或添加与安装相关的文档、产品版本和补丁更新。启用通知后，您可以更轻松地应用产品更新，如红帽客户门户网站中的更新可用，以便保持最新发布的发展版本，并附带最新的改进和修复。

### 先决条件

- 您有红帽客户门户网站帐户并登录。

### 流程

1. 在红帽客户门户网站窗口的右上角，点击您的资料名称并点击 **通知**。
2. 选择 **通知** 选项卡，然后单击 **管理通知**。
3. 在 **Follow** 旁边，从下拉菜单中选择 **Products**，然后从出现的下一下拉菜单中选择 **Red Hat Process Automation Manager**。
4. 点击 **Save Notification** 完成。

您可以根据需要为任何其他产品添加通知。



## 第 64 章 将补丁更新和次要发行升级应用到 RED HAT PROCESS AUTOMATION MANAGER

Red Hat Process Automation Manager 的补丁更新和新的次要版本中通常会提供自动更新，以帮助更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE 服务器和无头进程自动化管理器控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）都会作为新的工件发布，且您必须重新安装它们以应用更新。

您可以使用相同的自动更新工具将补丁更新和次要发行升级应用到 Red Hat Process Automation Manager 7.13。Red Hat Process Automation Manager 的补丁更新（如从版本 7.13 更新至 7.13.5）包括最新的安全更新和程序错误修复。Red Hat Process Automation Manager 的次发行版本升级，如从版本 7.12.x 升级到 7.13，包括功能增强、安全更新和程序错误修复。



### 注意

只有 Red Hat Process Automation Manager 的更新包含在 Red Hat Process Automation Manager 更新工具中。红帽 JBoss EAP 的更新必须使用红帽 JBoss EAP 补丁发布来应用。有关红帽 JBoss EAP 修补的更多信息，请参阅红帽 [JBoss EAP 修补和升级指南](#)。

### 先决条件

- 您的 Red Hat Process Automation Manager 和 KIE 服务器实例没有运行。在运行 Red Hat Process Automation Manager 或 KIE 服务器实例时，不要应用更新。

### 流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉列表中选择产品和版本。

如果您要升级到 Red Hat Process Automation Manager 的新次版本，如从版本 7.12.x 升级到 7.13，请首先将最新的补丁更新应用到 Red Hat Process Automation Manager 的当前版本，然后再次按照以下步骤升级到新的次版本。

2. 单击 Patches，下载 Red Hat Process Automation Manager [VERSION] Update Tool，并将下载的 rhpam-\$VERSION-update.zip 文件提取到临时目录。

这个版本工具会自动更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE 服务器和无头流程自动化管理器控制器。使用这个更新工具，首先应用更新，然后安装与 Red Hat Process Automation Manager 发行版本相关的其他更新或新的发行工件。

3.

如果要保留由更新工具更新的任何文件，请导航到提取的 `rhcam-$VERSION-update` 文件夹，打开 `blacklist.txt` 文件，并将相对路径添加到您不想要更新的文件。

当在 `blacklist.txt` 文件中列出文件时，更新脚本不会将该文件替换为新版本，而是将该文件保留到位，并在同一位置将新版本添加到 `.new` 后缀。如果您的块设备不再被分发，更新工具会创建一个带有 `.removed` 后缀的空标志文件。然后，您可以选择手动保留、合并或删除这些新文件。

在 `blacklist.txt` 文件中排除的文件示例：

```
WEB-INF/web.xml // Custom file
styles/base.css // Obsolete custom file kept for record
```

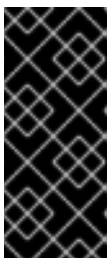
更新后阻止的文件目录的内容：

```
$ ls WEB-INF
web.xml web.xml.new
```

```
$ ls styles
base.css base.css.removed
```

4.

在命令终端中，导航到您提取 `rhcam-$VERSION-update.zip` 文件的临时目录，并以以下格式运行 `apply-updates` 脚本：



### 重要

在应用更新前，请确保您的 Red Hat Process Automation Manager 和 KIE 服务器实例不会运行。在运行 Red Hat Process Automation Manager 或 KIE 服务器实例时，不要应用更新。

在 Linux 或基于 Unix 的系统中：

```
$./apply-updates.sh $DISTRO_PATH $DISTRO_TYPE
```

在 Windows 中：

```
$.\apply-updates.bat $DISTRO_PATH $DISTRO_TYPE
```

`$DISTRO_PATH` 部分是相关分发目录的路径，而 `$DISTRO_TYPE` 部分是您使用此更新进行更新的发行类型。

Red Hat Process Automation Manager 更新工具支持以下分布类型：

- **RHPAM-business-central-eap7-deployable: 更新 Business Central(business-central.war)**
- **RHPAM-kie-server-ee8 : 更新 KIE 服务器(kie-server.war)**
- **RHPAM-kie-server-jws: 更新 KIE Server on Red Hat JBoss Web Server(kie-server.war)**
- **RHPAM-controller-ee7 : 更新无头进程自动化管理器控制器(controller.war)**
- **RHPAM-controller-jws: 更新 Red Hat JBoss Web Server(controller.war)上的无头进程自动化管理器控制器**

适用于 Red Hat JBoss EAP 上完整红帽流程自动化管理器分发的 Business Central 和 KIE 服务器示例：

```
$./apply-updates.sh ~EAP_HOME/standalone/deployments/business-central.war
rhpam-business-central-eap7-deployable
```

```
$./apply-updates.sh ~EAP_HOME/standalone/deployments/kie-server.war rhpam-kie-
server-ee8
```

使用，到无头进程 Automation Manager 控制器示例：

```
$./apply-updates.sh ~EAP_HOME/standalone/deployments/controller.war rhpam-
controller-ee7
```

更新脚本会在解压的 `rhpam-$VERSION-update` 文件夹中创建一个 备份 文件夹，其中包含指定分发的副本，然后继续更新。

5.

更新工具完成后，返回到红帽客户门户网站的 **Software Downloads** 页面，其中包括您下载

更新工具，并安装与 Red Hat Process Automation Manager 发行版本相关的其他更新或新的发行工件。

对于已在红帽流程自动化管理器分发中已存在的文件，如决策引擎或其他附加组件的 .jar 文件，将该文件的现有版本替换为红帽客户门户网站中的新版本。

6.

如果您使用独立的 Red Hat Process Automation Manager 7.13.5 Maven 存储库工件 (rhpam-7.13.5-maven-repository.zip)，如 air-gap 环境中，下载 Red Hat Process Automation Manager 7.13.5 Maven 存储库，并将下载的 rhpam-7.13.5-maven-repository.zip 文件提取到现有的 ~/maven-repository 目录，以更新相关内容。

**Maven 存储库更新示例：**

```
$ unzip -o rhpam-7.13.5-maven-repository.zip 'rhba-7.13.5.GA-maven-repository/maven-repository/*' -d /tmp/rhbaMavenRepoUpdate

$ mv /tmp/rhbaMavenRepoUpdate/rhba-7.13.5.GA-maven-repository/maven-repository/
$REPO_PATH/
```



**注意**

在完成更新后，您可以删除 /tmp/rhbaMavenRepoUpdate 文件夹。

7.

可选：如果您要使用基于属性的用户存储更改 Red Hat Process Automation Manager，请完成以下步骤：

a.

进入 \$JBOSS\_HOME 目录并运行以下命令之一：

- 

在 Linux 或基于 Unix 的系统中：

```
$./bin/standalone.sh --admin-only -c standalone-full.xml
```

```
$./bin/jboss-cli.sh --connect --file=rhpam-$VERSION-update/elytron/add-kie-fs-realm.cli
```

- 

在 Windows 中：

```
$./bin/standalone.bat --admin-only -c standalone-full.xml
```

-

```
$./bin/jboss-cli.bat --connect --file=rhpam-$VERSION-update/elytron/add-kie-fs-realm.cli
```

b.

运行以下命令：

•

在 Linux 或基于 Unix 的系统中：

```
$./bin/elytron-tool.sh filesystem-realm --users-file standalone/configuration/application-users.properties --roles-file standalone/configuration/application-roles.properties --output-location standalone/configuration/kie-fs-realm-users --filesystem-realm-name kie-fs-realm-users
```

•

在 Windows 中：

```
$./bin/elytron-tool.bat filesystem-realm --users-file standalone/configuration/application-users.properties --roles-file standalone/configuration/application-roles.properties --output-location standalone/configuration/kie-fs-realm-users --filesystem-realm-name kie-fs-realm-users
```

c.

导航到您提取 `rhpam-$VERSION-update.zip` 文件的目录，并运行以下命令来应用 `kie-fs-realm` 补丁之一：

•

在 Linux 或基于 Unix 的系统中：

```
$./elytron/kie-fs-realm-patch.sh ~/$JBOSS_HOME/standalone/configuration/kie-fs-realm-users/
```

•

在 Windows 中：

```
$./elytron/kie-fs-realm-patch.bat ~/$JBOSS_HOME/standalone/configuration/kie-fs-realm-users/
```

8.

应用完所有相关更新后，启动 Red Hat Process Automation Manager 和 KIE 服务器，并登录到 Business Central。

9.

验证 Business Central 窗口中的所有项目数据都显示和准确，然后在 Business Central 窗口右上角存在且准确，点击您的资料名称并点击 About 来验证更新的产品版本号。

如果您在 **Business Central** 中遇到错误或发现任何缺少的数据，您可以在 `rhpm-$VERSION-update` 文件夹中 恢复备份 文件夹中的内容来恢复更新工具更改。您还可以在红帽客户门户网站中重新安装之前版本的 **Red Hat Process Automation Manager** 中的相关发行工件。恢复之前的分发后，您可以重新尝试运行更新。

---

## 第 65 章 其他资源

- [规划 Red Hat Process Automation Manager 安装](#)
- [修补和升级 Red Hat Process Automation Manager 7.10](#)
- [Red Hat Process Automation Manager 7.13 发行注记](#)

## 附录 C. 版本信息

文档最新更新于 2024 年 3 月 14 日星期四。



## 附录 D. 联系信息

Red Hat Process Automation Manager 文档团队 : [brms-docs@redhat.com](mailto:brms-docs@redhat.com)