



Red Hat Process Automation Manager 7.12

安装和配置红帽流程自动化管理器

法律通告

Copyright © 2023 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 Red Hat JBoss EAP 集群环境中安装 Red Hat Process Automation Manager 组件，在 JBoss Web Server 上及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 MANAGER 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 参考实现	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. INTEGRATING RED HAT FUSE	17
8.3. INTEGRATING RED HAT SINGLE SIGN-ON	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. 使用安装程序使用流程自动化管理器控制器配置 KIE 服务器	27
第 14 章 从 ZIP 文件安装 RED HAT PROCESS AUTOMATION MANAGER	29
14.1. 从 ZIP 文件安装 BUSINESS CENTRAL	29
14.2. 从 ZIP 文件安装 KIE 服务器	29
14.3. 为 KIE 服务器配置 JDBC 数据源	30
14.4. 创建用户	32
14.5. 配置 KIE 服务器以连接到 BUSINESS CENTRAL	34
14.6. 线程效率	37
第 15 章 使用密钥存储保护密码	38

第 16 章 将 SSH 配置为使用 RSA	40
第 17 章 验证 RED HAT PROCESS AUTOMATION MANAGER 安装	41
第 18 章 运行 RED HAT PROCESS AUTOMATION MANAGER	42
第 19 章 安装并运行无头进程 AUTOMATION MANAGER 控制器	43
19.1. 安装无头进程 AUTOMATION MANAGER 控制器	43
19.2. 运行无头进程 AUTOMATION MANAGER 控制器	45
第 20 章 配置和运行独立 BUSINESS CENTRAL	47
第 21 章 RED HAT PROCESS AUTOMATION MANAGER 的 MAVEN 设置和存储库	49
21.1. 为 RED HAT PROCESS AUTOMATION MANAGER 添加 MAVEN 依赖项	49
21.2. 为 BUSINESS CENTRAL 和 KIE 服务器配置外部 MAVEN 存储库	52
21.3. 准备 MAVEN 镜像存储库以供离线使用	53
第 22 章 从 GIT 存储库导入项目	56
第 23 章 自定义 BUSINESS CENTRAL 的品牌	57
23.1. 自定义 BUSINESS CENTRAL 登录页面	57
23.2. 自定义 BUSINESS CENTRAL 应用程序标头	57
第 24 章 集成 LDAP 和 SSL	58
附录 A. BUSINESS CENTRAL 系统属性	59
第 25 章 其他资源	64
第 26 章 后续步骤	65
部分 III. 在 RED HAT JBOSS EAP 集群环境中安装和配置 RED HAT PROCESS AUTOMATION MANAGER ..	66
第 27 章 RED HAT PROCESS AUTOMATION MANAGER 集群	67
第 28 章 开发（授权）环境中的 RED HAT PROCESS AUTOMATION MANAGER 集群	68
28.1. 安装和配置 RED HAT DATA GRID	68
28.2. 下载和配置 AMQ BROKER	69
28.3. 配置 NFS 版本 4 服务器	70
28.4. 下载并提取 RED HAT JBOSS EAP 7.4 和 RED HAT PROCESS AUTOMATION MANAGER	71
28.5. 在集群中配置和运行 BUSINESS CENTRAL	71
28.6. 测试内部基础架构的高可用性(HA)	73
28.7. 验证 RED HAT PROCESS AUTOMATION MANAGER 集群	76
第 29 章 运行时环境中的 KIE 服务器集群	77
29.1. 下载并提取 RED HAT JBOSS EAP 7.4 和 KIE 服务器	77
29.2. 为 KIE 服务器配置并运行 RED HAT JBOSS EAP 7.4 集群	78
29.3. 使用无头进程 AUTOMATION MANAGER 控制器集群 KIE 服务器	81
29.4. 在 KIE 服务器节点上为 RUNNING 状态配置故障切换	82
第 30 章 安装和配置智能路由器	85
30.1. 使用智能路由器实现负载均衡 KIE 服务器实例	85
30.2. 为 TLS 支持配置 SMART ROUTER	87
30.3. 为端点身份验证配置智能路由器	88
30.4. 配置智能路由器行为	89
30.5. 使用配置文件配置智能路由器设置	91
第 31 章 配置 QUARTZ 计时器服务	93

第 32 章 其他资源	95
部分 IV. 在 RED HAT JBOSS WEB SERVER 上安装和配置 RED HAT PROCESS AUTOMATION MANAGER	96
第 33 章 关于 RED HAT PROCESS AUTOMATION MANAGER	97
33.1. RED HAT PROCESS AUTOMATION MANAGER 组件	97
33.2. RED HAT PROCESS AUTOMATION MANAGER 角色和用户	98
第 34 章 下载 RED HAT PROCESS AUTOMATION MANAGER 安装文件	100
第 35 章 使用 RED HAT PROCESS AUTOMATION MANAGER 安装程序	101
35.1. 在互动模式中使用安装程序	101
35.2. 在 CLI 模式中使用安装程序	103
第 36 章 KIE SERVER ZIP 文件安装和配置	106
36.1. 从 ZIP 文件安装 KIE 服务器	106
36.2. 配置 JDBC WEB 服务器数据源	107
第 37 章 使用密钥存储保护密码	112
第 38 章 验证 KIE 服务器安装	114
第 39 章 下载并安装无头进程 AUTOMATION MANAGER 控制器	115
第 40 章 配置和运行独立 BUSINESS CENTRAL	117
第 41 章 RED HAT PROCESS AUTOMATION MANAGER 的 MAVEN 设置和存储库	119
41.1. 使用项目配置文件(POM.XML)配置 MAVEN	119
41.2. 修改 MAVEN 设置文件	119
41.3. 为 RED HAT PROCESS AUTOMATION MANAGER 添加 MAVEN 依赖项	119
41.4. 准备 MAVEN 镜像存储库以供离线使用	123
第 42 章 从 GIT 存储库导入项目	126
第 43 章 集成 LDAP 和 SSL	127
附录 B. BUSINESS CENTRAL 系统属性	128
部分 V. 在 IBM WEBSHERE APPLICATION SERVER 上安装和配置 KIE 服务器	133
第 44 章 KIE SERVER	134
第 45 章 IBM WEBSHERE APPLICATION SERVER	135
第 46 章 安装并运行 IBM WEBSHERE APPLICATION SERVER	136
第 47 章 为 KIE 服务器配置 IBM WEBSHERE APPLICATION SERVER	137
47.1. 启用管理安全性	137
47.2. 在 IBM WEBSHERE APPLICATION SERVER 中配置 JDBC 数据源	138
47.3. 配置 JAVA 消息服务(JMS)	139
47.4. 在 IBM WEBSHERE APPLICATION SERVER 中设置系统属性	142
47.5. 停止并重启 IBM WEBSHERE APPLICATION SERVER	145
第 48 章 使用 IBM WEBSHERE APPLICATION SERVER 安装并运行 KIE SERVER	147
48.1. 创建 KIE SERVER 组和角色	148
48.2. 映射 KIE SERVER 组和角色	148
48.3. 为 KIE 服务器配置类加载	149
48.4. 验证安装	149

第 49 章 使用 IBM WEBSHERE APPLICATION SERVER 安装并运行无头流程 AUTOMATION MANAGER 控制器	150
49.1. 为无头进程 AUTOMATION MANAGER 控制器配置类加载	151
49.2. 为无头进程 AUTOMATION MANAGER 控制器设置系统属性	151
49.3. 映射无头进程 AUTOMATION MANAGER 控制器组和角色	152
49.4. 验证安装	153
第 50 章 在 IBM WEBSHERE APPLICATION SERVER 中配置嵌入式流程引擎和决策引擎	154
第 51 章 使用密钥存储保护密码	158
第 52 章 后续步骤	160
部分 VI. 在 ORACLE WEBLOGIC SERVER 上安装和配置 KIE 服务器	161
第 53 章 KIE SERVER	162
第 54 章 ORACLE WEBLOGIC SERVER	163
第 55 章 安装并运行 ORACLE WEBLOGIC SERVER	164
第 56 章 为 KIE 服务器配置 ORACLE WEBLOGIC 服务器	166
56.1. 配置 KIE SERVER 组和用户	166
56.2. 在 ORACLE WEBLOGIC 服务器中配置 JDBC 数据源	166
56.3. 配置 JAVA 消息服务(JMS)	168
56.4. 在 ORACLE WEBLOGIC SERVER 中设置系统属性	171
56.5. 停止并重启 ORACLE WEBLOGIC SERVER	172
第 57 章 使用 ORACLE WEBLOGIC SERVER 安装 KIE 服务器	174
57.1. 验证 ORACLE WEBLOGIC SERVER 上的 KIE 服务器安装	174
第 58 章 使用 ORACLE WEBLOGIC SERVER 安装并运行无头进程 AUTOMATION MANAGER 控制器	176
58.1. 为无头进程 AUTOMATION MANAGER 控制器设置系统属性	176
58.2. 验证安装	177
第 59 章 在 ORACLE WEBLOGIC SERVER 中配置嵌入式流程引擎和决策引擎	179
第 60 章 使用密钥存储保护密码	183
第 61 章 后续步骤	185
部分 VII. 修补并升级 RED HAT PROCESS AUTOMATION MANAGER 7.12	186
第 62 章 RED HAT PROCESS AUTOMATION MANAGER 版本	187
第 63 章 在红帽客户门户网站中启用 RED HAT PROCESS AUTOMATION MANAGER 通知	188
第 64 章 将补丁更新和次版本的升级应用到 RED HAT PROCESS AUTOMATION MANAGER	189
第 65 章 其他资源	193
附录 C. 版本控制信息	194
附录 D. 联系信息	195

前言

作为开发者或系统管理员，您可以在 Red Hat JBoss EAP 集群环境中安装 Red Hat Process Automation Manager 组件，在 JBoss Web Server 和其他支持的环境中安装 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、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 信息](#)。

部分 I. 规划 RED HAT PROCESS AUTOMATION MANAGER 安装

作为系统管理员，您有几个安装 Red Hat Process Automation Manager 的选项。

第 1 章 RED HAT PROCESS AUTOMATION MANAGER 版本

Red Hat Process Automation Manager 版本指定为数字 **Major.Minor.Patch** 格式，如 **7.5.1**。在本例中，主发行版本是 **7.x.x**，次发行版本为 **7.5.x**，补丁版本为 **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发行版本移到另一个主发行版本（如从 Red Hat JBoss BPM Suite 6.4.x 到 Red Hat Process Automation Manager 7.0）时，通常会需要数据迁移。自动化迁移工具通常随 Red Hat Process Automation Manager 的新主要版本提供，以协助迁移，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。

有关迁移说明示例，[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 的次发行版本包括功能增强、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 移到 7.6。自动更新工具通常由 Red Hat Process Automation Manager 的补丁更新和新的次版本提供，以便更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE Server 和无头流程 Automation Manager 控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）作为每个次发行版本的新工件发布，您必须重新安装它们以应用更新。

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

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该产品的所有以前发布的补丁更新，因此您不需要逐步应用每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，为了获得最佳红帽流程自动化管理器性能，请在产品更新可用时应用它们。有时，红帽可能会在现有产品普通更新周期外发布非计划补丁更新。它们可能包括 Red Hat 全球支持服务(GSS)提供的安全性或其他更新来修复特定问题，且可能不是累积更新。

自动更新工具通常由 Red Hat Process Automation Manager 的补丁更新和新的次版本提供，以便更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE Server 和无头流程 Automation Manager 控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）作为每个次发行版本的新工件发布，您必须重新安装它们以应用更新。

为确保在发行版本之间实现最佳过渡，并保持红帽流程自动化管理器最新发布，在红帽客户门户网站中提供新产品版本及 Red Hat Process Automation Manager 的更新。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

Red Hat Process Automation Manager 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) 中安装 Business Central。商业中心也作为独立 JAR 文件提供。您可以使用 Business Central 独立 JAR 文件来运行 Business Central，而不必将它部署到应用服务器。
- KIE Server 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计划问题。您可以在 Red Hat JBoss EAP 实例、Red Hat JBoss EAP 集群中、在 OpenShift 上安装 KIE 服务器，在一个 Oracle WebLogic 服务器实例、IBM WebSphere Application Server 实例中，或作为 Spring Boot 应用程序的一部分。您可以将 KIE 服务器配置为以受管或非受管模式运行。如果 KIE Server 是非受管服务器，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，流程自动化管理器控制器管理 KIE 服务器配置，并与 Process Automation Manager 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 Execution Server 页面创建和维护 KIE 容器。但是，如果您没有安装 Business Central，您可以安装无头进程 Automation Manager 控制器，并使用 REST API 或 KIE Server Java 客户端 API 与之交互。

- 红帽构建的 OptaPlanner 已在 Business Central 和 KIE 服务器中集成。它是可嵌入的轻量级规划引擎，可优化规划问题。红帽构建的 OptaPlanner 帮助 Java 编程人员有效地解决规划问题，并将优化风格和元化与分数计算相结合。

第 3 章 RED HAT PROCESS AUTOMATION MANAGER 安装平台

每个 Red Hat Process Automation Manager 版本都在各种操作系统、JVM、Web 浏览器和数据库组合上认证。根据您的订阅协议，红帽为物理和虚拟环境中的受支持和测试配置提供生产和开发支持。有关支持的配置和版本号的详情，请查看以下页面：

- [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 7.12 可用于以下应用平台：

- 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

红帽 JBoss 企业应用平台（红帽 JBoss EAP）是 Java Enterprise Edition 7 (Java EE 7)完整和 Web 配置文件规范的已认证实施。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。它还允许用户使用红帽 JBoss EAP 提供的各种 API 和服务编写、部署和运行应用。

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，您应该在不同的实例中安装它们。

3.2. RED HAT JBOSS WEB SERVER

Red Hat JBoss Web Server 是一个基于 Tomcat 的中型和大型应用程序设计的企业级 Web 服务器。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 服务器和无头流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 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 的组件被部署为单独的 OpenShift pod。您可以单独扩展和缩减每个 pod，从而根据特定组件所需的数量或多个容器提供。您可以使用标准的 OpenShift 方法管理容器集并平衡负载。

您可以在 OpenShift 上安装 Business Central 和 KIE Server。

3.4. ORACLE WEBLOGIC SERVER

Oracle WebLogic Server 是一个 Java EE 应用服务器，它提供一组标准的 API 来创建分布式 Java 应用程序，可访问各种服务，如数据库、消息传递服务以及到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 Server Java 客户端 API 与 KIE 服务器交互。您还可以使用独立的 Business Central 与 KIE 服务器交互。

3.5. IBM WEBSPHERE APPLICATION SERVER

IBM WebSphere Application Server 是一个灵活的安全 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，自版本 8.5.5.6 起完全兼容 Java EE 7。

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

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

您可以在 Red Hat Process Automation Manager 中使用 Git 存储库和 Apache Maven。

4.1. GIT 存储库

Git 存储库在内部使用，用于存储在编写环境中创建的所有进程、规则和其他工件。Git 是一个分布式版本控制系统。它将修订版本作为提交对象实施。当您将更改提交到存储库时，会在 Git 存储库中创建新的提交对象。当您在 Business Central 中创建项目时，会将其添加到连接到 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 空格。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 Java 应用程序开发中用于构建和管理软件项目的分布式构建自动化工具。您可以使用 Maven 构建、发布和部署红帽流程自动化管理器项目。Maven 提供以下优点：

- 构建过程比较简单，它在项目间实施统一构建系统。
- 在编译时，项目所需的所有 JAR 文件都提供。
- 配置了适当的项目结构。
- 依赖项和版本得到很好的管理。
- 不需要额外的构建处理，因为 Maven 构建会输出到许多预定义的类型，如 JAR 和 WAR。

Maven 使用存储库来存储 Java 库、插件和其他构建构件。这些仓库可以是本地或远程存储库。Red Hat Process Automation Manager 维护本地和远程 maven 存储库，您可以将其添加到项目中以访问规则、进程、事件和其他项目依赖项。在构建项目和架构类型时，Maven 会动态从本地或远程存储库检索 Java 库和 Maven 插件。这样做可促进项目间的依赖关系共享和重复使用。

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

通过 Red Hat Process Automation Manager，您可以设置开发环境来开发业务应用程序、运行这些应用程序的运行时环境来支持决策。

- **开发环境**：Typly 包括一个 Business Central 安装和至少一个 KIE 服务器安装。您可以使用 Business Central 设计决策和其他工件，您可以使用 KIE Server 执行并测试您创建的工件。
- **运行时环境**：使用或不使用 Business Central 的一个或多个 KIE 服务器实例。Business Central 有一个嵌入式流程 Automation Manager 控制器。如果安装了 Business Central，请使用 **Menu → Deploy → Execution servers** 页面创建和维护容器。如果要在不使用 Business Central 的情况下自动执行 KIE 服务器管理，您可以使用无头进程 Automation Manager 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组成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几率。当服务器出现故障时，另一台服务器通过提供故障服务器上的数据副本来填补差距。当失败的服务器再次上线时，它会恢复它在集群中的位置。



注意

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

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

要访问 Business Central 或 KIE Server，您必须创建用户并在服务器启动前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 服务器都在单一实例上运行，则对 Business Central 进行身份验证的用户也可以访问 KIE 服务器。

但是，如果 Business Central 和 KIE 服务器在不同实例上运行，则对 Business Central 进行身份验证的用户必须单独通过身份验证才能访问 KIE 服务器。例如，如果一个在 Business Central 上通过身份验证但未通过 KIE Server 进行身份验证的用户尝试查看或管理 Business Central 中的进程定义，则会在日志文件中记录 401 错误，以及从远程服务器加载数据的 **Invalid 凭证**。请联系您的系统管理员。信息出现在 Business Central 中。

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



注意

admin, analyst, developer, manager, process-admin, user, 和 rest-all 角色为 Business Central 保留。**kie-server** 角色为 KIE 服务器保留。因此，可用的角色可能会因 Business Central、KIE Server 还是安装这两者而异。

- **admin**：具有管理 **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。它们具有对应用程序所需的更改的完整访问权限。具有 **admin** 角色的用户可以访问 Red Hat Process Automation Manager 中的所有区域。
- **分析师**：具有 **分析者** 角色的用户有权访问所有高级别功能。他们可以对他们的项目进行建模和执行。但是，这些用户无法在 **Design → Projects** 视图中为空格或删除空格添加贡献者。带有 **analyst** 角色的用户没有访问 **Deploy → Execution Servers** 视图（面向管理员）的访问权限。但是，当这些用户访问 Library 透视图时，可以使用 **Deploy** 按钮。
- **开发人员**：具有 **developer** 角色的用户几乎可以访问所有的功能，并且可以管理规则、模型、流程、表单和仪表盘。他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）才会在具有 **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。这些用户通常对业务流程及其性能、业务指示器和其他业务相关报告相关的统计信息感兴趣。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。他们可完全访问业务流程、业务任务和执行错误。这些用户还可以查看业务报告，并可访问 Task Inbox 列表。
- **用户**：具有 **用户角色** 的用户可以处理 Task Inbox 列表，其中包含属于当前运行的进程的商业任务。具有此角色的用户可以查看进程和任务报告和管理进程。
- **rest-all**：拥有 **rest-all** 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 Server REST 功能。对于需要访问 Business Central 中的 **Manage** 和 **Track** 视图的用户，这个角色是必须的。

第 7 章 RED HAT PROCESS AUTOMATION MANAGER 参考实现

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 **Red Hat Process Automation Manager 7.12.0 Reference Implementations** 下载中，可在红帽客户门户网站中的 [Red Hat Process Automation Manager Software Downloads](#) 页面中提供。

员工故障参考实施

通过员工对参考实施的实施，您可以创建一个应用程序，使员工能够把员工转移到一个机构中的不同位置。例如，您可以使用应用程序在国内、保护在多个位置间转移转移，或在工作程序之间的配配线上切换。

vehicle 路由计划参考实施

通过 vehicle 路由计划实施实施，您可以创建一个应用程序来解决真实映射、路线和车辆以特定容量为位置解决路由计划问题的应用程序。如需更多信息，请参阅参考实施下载中包含的 vehicle 路由 ZIP 文件中的 README 文件。

学校时间表参考实现

通过按时序参考实施，您可以在 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，并通过考虑学员和教师限制来避免冲突。

高度可用的事件驱动决策参考实施

通过高可用性事件驱动的决策参考实施，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开发的规则）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 与其它产品和组件（如 Spring Boot、Red Hat Fuse 和 Red Hat Single Sign-On）集成。

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

Spring Framework 是一个 Java 平台，它为开发 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，其中包含您可以在 Spring Boot 项目中包含的一组依赖项描述符。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、独立于 UI 的逻辑分组，提供某些业务功能。业务应用程序基于 Spring Boot 发起者。它们通常单独部署，可以单独进行版本控制。完整的业务应用程序使域能够实现特定的业务目标，例如订购管理或调整管理。

在 [商业应用程序](#) 网站上，您可以创建流程自动化管理器、决策管理器或红帽构建 OptaPlanner 业务应用程序。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中。

8.2. INTEGRATING RED HAT FUSE

Fuse 是作为灵活整合解决方案一部分的分布式云原生集成平台。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 上安装红帽 Fuse，然后在该容器中安装和配置红帽流程自动化管理器。

8.3. INTEGRATING RED HAT SINGLE SIGN-ON

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

当您将 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为互联网上的不同红帽流程自动化管理器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 Server 上安装和配置 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 Red Hat Process Automation Manager 集成](#)
- [将 Red Hat Process Automation Manager 与 Red Hat Single Sign-On 集成](#)

红帽构建的 OptaPlanner

- [使用 Red Hat Process Automation Manager 开发 Solvers](#)
- [在 Red Hat OpenShift Container Platform 上使用决策引擎实施高度可用的事件驱动决策](#)

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](#)
- [在 Red Hat OpenShift Container Platform 上使用决策引擎实施高度可用的事件驱动决策](#)

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

- 您已查看了 [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 Red Hat 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。它旨在使业务和 IT 用户能够更有效地协作，从而轻松、快速地更改业务应用程序。

Red Hat Process Automation Manager 由 Business Central 和 KIE 服务器组成。

- KIE Server 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计划问题。KIE 服务器为业务资产提供运行时环境，并访问资产存储库中存储的数据（知识存储）。
- Business Central 是您创建和管理 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。商业中心也作为独立 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)完整和 Web 配置文件规格的经过认证的实现。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。它还允许用户使用红帽 JBoss EAP 提供的各种 API 和服务编写、部署和运行应用。

本文档中的说明解释了如何在 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 Server 上安装和配置 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 Server，您必须创建用户并在服务器启动前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 服务器都在单一实例上运行，则对 Business Central 进行身份验证的用户也可以访问 KIE 服务器。

但是，如果 Business Central 和 KIE 服务器在不同实例上运行，则对 Business Central 进行身份验证的用户必须单独通过身份验证才能访问 KIE 服务器。例如，如果一个在 Business Central 上通过身份验证但未通过 KIE Server 进行身份验证的用户尝试查看或管理 Business Central 中的进程定义，则会在日志文件中记录 401 错误，以及从远程服务器加载数据的 **Invalid 凭证**。请联系您的系统管理员。信息出现在 Business Central 中。

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



注意

admin, analyst, developer, manager, process-admin, user 和 **rest-all** 角色为 Business Central 保留。**kie-server** 角色为 KIE 服务器保留。因此，可用的角色可能会因 Business Central、KIE Server 还是安装这两者而异。

- **admin**：具有管理 **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。它们具有对应用程序所需的更改的完整访问权限。具有 **admin** 角色的用户可以访问 Red Hat Process Automation Manager 中的所有区域。
- **分析师**：具有 **分析者** 角色的用户有权访问所有高级别功能。他们可以对他们的项目进行建模和执行。但是，这些用户无法在 **Design → Projects** 视图中为空格或删除空格添加贡献者。带有 **analyst** 角色的用户没有访问 **Deploy → Execution Servers** 视图（面向管理员）的访问权限。但是，当这些用户访问 Library 透视图时，可以使用 **Deploy** 按钮。
- **开发人员**：具有 **developer** 角色的用户几乎可以访问所有的功能，并且可以管理规则、模型、流程、表单和仪表盘。他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）才会在具有 **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。这些用户通常对业务流程及其性能、业务指示器和其他业务相关报告相关的统计信息感兴趣。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。他们可完全访问业务流程、业务任务和执行错误。这些用户还可以查看业务报告，并可访问 Task Inbox 列表。
- **用户**：具有 **用户角色** 的用户可以处理 Task Inbox 列表，其中包含属于当前运行的进程的商业任务。具有此角色的用户可以查看进程和任务报告和管理进程。
- **rest-all**：拥有 **rest-all** 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 Server REST 功能。对于需要访问 Business Central 中的 **Manage** 和 **Track** 视图的用户，这个角色是必须的。

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 Server 部署 ZIP 文件。如果您要运行 Business Central 而不将其部署到应用服务器，请下载 Business Central Standalone JAR 文件。

下载满足您的环境和安装要求的 Red Hat Process Automation Manager 发行版本。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - **产品：** 流程自动化管理器
 - **Version:** 7.12
2. 根据您首选的安装方法下载以下产品发行版之一：



注意

您只需要下载其中一个发行版。

- 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 7.12.0 Installer(rhpam-installer-7.12.0.jar)**。安装程序图形用户界面指导您完成安装过程。
- 如果要使用可部署的 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：
 - **Red Hat Process Automation Manager 7.12.0 KIE Server for All Supported EE8 Containers (rhpam-7.12.0-kie-server-ee8.zip)**
 - **Red Hat Process Automation Manager 7.12.0 Business Central Deployable for EAP 7 (rhpam-7.12.0-business-central-central-eap7-deployable.zip)**
 - **Red Hat Process Automation Manager 7.12.0 Add Ons(rhpam-7.12.0-add-ons.zip)**
- 要运行 Business Central，无需将它部署到应用程序服务器，请下载 **Red Hat Process Automation Manager 7.12.0 Business Central Standalone (rhpam-7.12.0-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。



注意

Red Hat Process Automation Manager JAR 文件安装程序 *不支持* 由 yum 或 RPM 软件包管理器安装的 Red Hat JBoss EAP 分发版。如果要在这种类型的 Red Hat JBoss EAP 安装中安装 Red Hat Process Automation Manager，请下载 **Red Hat Process Automation Manager 7.12 Deployable for Red Hat JBoss EAP 7.4** 文件，并遵循 [第 14 章 从 ZIP 文件安装 Red Hat Process Automation Manager](#) 中的步骤。



注意

因为 IBM JDK 无法使用其他 JDK 上生成的密钥存储，所以您无法使用另一个 JDK 上生成的密钥存储在 IBM JDK 上运行的现有 Red Hat JBoss EAP 安装中安装。

后续步骤：

按照以下部分之一的说明操作：

- [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 “使用安装程序使用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。

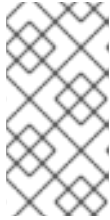


注意

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 7.12.0 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](#)。
- 提供了备份的 Red Hat JBoss EAP 7.4 服务器安装。
- 授予足够的用户权限来完成安装。
- 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 旨在与 UTF-8 编码一起工作。如果底层 JVM 使用了不同的编码系统，则可能会出现意外的错误。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，请参阅 [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 文件的目录，然后输入以下命令：

```
java -jar rhpam-installer-7.12.0.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据。要防止此要求，请在安装命令中添加 `izpack.mode=privileged` 选项：

```
java -Dizpack.mode=privileged -jar  
rhpam-installer-7.12.0.jar
```

另外，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-7.12.0.jar
```

图形安装程序显示启动屏幕和许可证协议页面。

2. 单击 **I accept the terms of this license agreement**，然后单击 **Next**。
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

您可以在同一个服务器上安装 Business Central 和 KIE 服务器。但是，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。为此，请运行安装程序两次。

5. 创建用户，**然后单击下一步**。默认情况下，如果您在同一容器中安装 Business Central 和 KIE Server，则新用户将获得 **admin**、**kie-server** 和 **rest-all** 角色。如果您只安装 KIE Server，则会给出 **kie-server** 角色。要选择另一个角色，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 **admin** 的用户。

密码必须至少有 8 个字符，且必须至少包含一个数字，以及一个非字母数字字符，但不能包括 & (ampersand)。

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

6. 在 **Installation Overview** 页面中，查看您要安装的组件并点 **Next** 开始安装。
7. 安装完成后，单击 **下一步**。
8. 如果安装了 KIE Server，则 **组件安装** 下会显示 **Configure Runtime** 步骤。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 **Perform advanced configuration**，您可以选择配置数据库设置或自定义某些 KIE Server 选项。
9. 如果您选择了 **Customize database settings**，在 **JDBC Drive Configuration** 页面中指定数据源 JDBC 驱动程序供应商，请选择一个或多个驱动程序 JAR 文件，然后单击 **Next**。
数据源是一种对象，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，以确保服务器与指定数据库之间的正确数据交换。
10. 如果您选择了 **Customize KIE Server properties**，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 Server 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。
 - 取消选择您要禁用的任何 KIE 服务器功能。
11. 点 **Next** 配置运行时环境。
12. 当在屏幕的顶部出现 **Processing finished** 时，点 **Next** 来完成安装。
13. 可选：点 **Generate Installation Script and Properties File** 将安装数据保存在 XML 文件中，然后点 **Done**。
安装程序生成两个文件。**auto.xml** 文件自动执行将来的安装，**auto.xml.variables** 文件存储用户密码和其他敏感变量。使用 **auto.xml** 文件在多个具有相同类型的服务器以及与原始安装相同的配置的情况下重复 Red Hat Process Automation Manager 安装。如有必要，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-7.12.0.jar <path-to-auto.xml-file>
```

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



注意

如果使用 Microsoft SQL Server，请确保已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。建议的配置是通过输入以下语句来打开

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 7.12.0 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](#)。
- 提供了备份的 Red Hat JBoss EAP 7.4 服务器安装。
- 授予足够的用户权限来完成安装。
- 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 旨在与 UTF-8 编码一起工作。如果底层 JVM 使用了不同的编码系统，则可能会出现意外的错误。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，请参阅 [附录 A, Business Central 系统属性](#)。

流程

1. 在终端窗口中，进入您下载安装程序文件的目录，并输入以下命令：

```
java -jar rhpam-installer-7.12.0.jar -console
```

命令行交互流程将启动并显示最终用户许可协议。

```
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。
安装程序将在提供的位置验证安装位置。输入 **1** 确认并继续。



注意

您可以在同一个服务器上安装 Business Central 和 KIE 服务器。但是，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。

4. 按照安装程序中的说明完成安装。



注意

当您创建用户名和密码时，请确保指定的用户名与角色或组群的任何已知标题没有冲突。例如，如果有一个名为 **admin** 的角色，则不应创建一个名为 **admin** 的用户。

密码必须至少有 8 个字符，且必须至少包含一个数字，以及一个非字母数字字符（不能包括字符 **&**）。

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

5. 安装完成后，您将看到这个信息：

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

6. 输入 **y** 创建包含安装数据的 XML 文件，或输入 **n** 来完成安装。如果输入 **y**，系统会提示您指定 XML 文件的路径。
7. 输入路径或按 Enter 键接受推荐的路径。
安装程序生成两个文件。**auto.xml** 文件自动执行将来的安装，**auto.xml.variables** 文件存储用户密码和其他敏感变量。在多个系统上使用 **auto.xml** 文件，在与原始安装相同的配置相同的服务器上轻松重复 Red Hat Process Automation Manager 安装。如有必要，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-7.12.0.jar <path-to-auto.xml-file>
```

8. 如果您只安装 Business Central，请重复这些步骤以在单独的服务器上安装 KIE Server。



注意

如果使用 Microsoft SQL Server，请确保已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。建议的配置是通过输入以下语句来打开 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. 使用安装程序使用流程自动化管理器控制器配置 KIE 服务器

KIE 服务器可由 Process Automation Manager 控制器管理，也可以被非受管管理。如果 KIE Server 是非受管服务器，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，流程自动化管理器控制器管理 KIE 服务器配置，并与 Process Automation Manager 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。

您可以在互动或 CLI 模式中使用安装程序来安装 Business Central 和 KIE 服务器，然后使用 Process Automation Manager 控制器配置 KIE 服务器。

先决条件

- 提供了两个带有备份的 Red Hat JBoss EAP 7.4 服务器安装的计算机。
- 授予足够的用户权限来完成安装。

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 “在交互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 **组件选择** 页面中，清除 **KIE Server** 框。
3. 完成 Business Central 安装。
4. 在第二台计算机上，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** 页面上，选择 **Perform Advanced Configuration**。
7. 选择 **Customize KIE Server 属性** 并单击下一步。
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 具有以下形式，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 Process Automation Manager 控制器现在是否已与 Business Central 集成，请转至 Business Central 中的 **Execution Servers** 页面，并确认您在 **REMOTE SERVERS** 下配置的 KIE 服务器。

第 14 章 从 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，一个用于 KIE Server）在不用安装程序的情况下安装红帽流程自动化管理器。



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 Automation Manager 控制器](#)。

14.1. 从 ZIP 文件安装 BUSINESS CENTRAL

Business Central 是您创建和管理 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

- 提供了一个受支持的 Red Hat JBoss EAP 安装版本 7.4。红帽 JBoss EAP 安装的基础目录称为 **EAP_HOME**。
- 授予足够的用户权限来完成安装。
- 下载以下文件，如 [第 12 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述：
rhpmam-7.12.0-business-central-eap7-deployable.zip

流程

1. 将 **rhpmam-7.12.0-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/rhpmam-7.12.0-business-central-eap7-deployable/jboss-eap-7.4** 目录的内容复制到 **EAP_HOME**。出现提示时，合并或替换文件。



警告

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

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，并访问资产存储库中存储的数据（知识存储）。您可以使用可部署资源 ZIP 文件来安装和配置 KIE 服务器。

先决条件

- 提供了一个受支持的 Red Hat JBoss EAP 安装版本 7.4。红帽 JBoss EAP 安装的基础目录称为 **EAP_HOME**。
- 授予足够的用户权限来完成安装。
- 下载以下文件，如 [第 12 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述：
rhpmam-7.12.0-kie-server-ee8.zip

流程

1. 将 **rhpmam-7.12.0-kie-server-ee8.zip** 归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/rhpmam-7.12.0-kie-server-ee8/kie-server.war** 目录复制到 **EAP_HOME/standalone/deployments/**。



警告

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

3. 将 **TEMP_DIR/rhpmam-7.12.0-kie-kie-server-ee8/rhpmam-7.12.0-kie-server-ee8/SecurityPolicy/** 复制到 **EAP_HOME/bin**。
4. 当提示覆盖文件时，点 **替换**。
5. 在 **EAP_HOME/standalone/deployments/** 目录中，创建一个名为 **kie-server.war.dodeploy** 的空文件。此文件确保服务器启动时自动部署 KIE 服务器。

注意

如果使用 Microsoft SQL Server，请确保已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。建议的配置是通过输入以下语句来打开 **ALLOW_SNAPSHOT_ISOLATION** 和 **READ_COMMITTED_SNAPSHOT**：

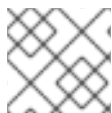
```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

14.3. 为 KIE 服务器配置 JDBC 数据源

数据源是一种对象，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，以确保服务器与指定数据库之间的正确数据交换。

通常，使用红帽流程自动化管理器的解决方案在单个事务中管理多个资源。JMS 用于异步作业、事件和计时器，例如：Red Hat Process Automation Manager 尽可能在数据源中有一个 XA 驱动程序，以确保数据原子性和结果一致。如果不同架构的事务代码存在于监听器内，或源自 jBPM 引擎提供的 hook，则还需要 XA 驱动程序。

除非您没有参与单个事务的多个资源，否则不要使用非 XA 数据源。



注意

对于生产环境，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

- 您要用于创建数据库连接的 JDBC 提供程序是在您要在其上部署 KIE 服务器的所有服务器上配置的，如 [Red Hat JBoss Enterprise Application Server Configuration Guide](#) 的 "Creating Datasources" 和 "JDBC Drivers" 部分所述。
- Red Hat Process Automation Manager 7.12.0 Add Ons([rhpam-7.12.0-add-ons.zip](#))文件是从红帽客户门户网站中的 [Software Downloads](#) 页面下载。

流程

1. 完成以下步骤以准备数据库：
 - a. 将 **rhpam-7.12.0-add-ons.zip** 提取到临时目录中，如 **TEMP_DIR**。
 - b. 提取 **TEMP_DIR/rhpam-7.12.0-migration-tool.zip**。
 - c. 将您的当前目录更改为 **TEMP_DIR/rhpam-7.12.0-migration-tool/ddl-scripts** 目录。此目录包含多种数据库类型的 DDL 脚本。
 - d. 将数据库类型的 DDL 脚本导入到您要使用的数据库中。

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

如果您将 PostgreSQL 或 Oracle 与 Spring Boot 结合使用，您必须导入相应的 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，其中包含标有 **@LOB** 的实体属性的自动递增整数值(OID)列。要使用其他二进制列类型，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.useBytea=true** 标志。Red Hat Process Automation Manager 不提供从基于 OID 变为基于 BYTEA 模式的迁移工具。

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>** 是数据库的休眠。



注意

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

- DB2：`org.hibernate.dialect.DB2Dialect`
- MSSQL：`org.hibernate.dialect.SQLServer2012Dialect`
- 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.PostgresPlusDialect`
- archive：`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，然后才能登录到 Business Central 或 KIE Server，您必须创建用户。本节演示了如何使用 `admin`、`rest-all` 和 `kie-server` 角色和具

有 **kie-server** 角色的 KIE Server 用户创建 Business Central 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 将用户数据存储为一组属性或一组文件。基于文件的存储提供多种额外功能，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

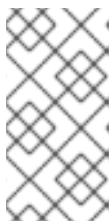
- Red Hat Process Automation Manager 安装在 Red Hat JBoss EAP 安装(**EAP_HOME**)的基础目录中。

流程

- 可选：要将 Red Hat Process Automation Manager 从使用基于属性的用户存储改为基于文件的用户存储，请完成以下步骤：
 - 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
          
```
 - 打开每个 **kie-fs-realm-users gcloud/<USER>.xml** 文件，其中 **<USER>** 是 Red Hat Process Automation Manager 用户。
 - 将 **<attribute name="roles" value=** 替换为 **<attribute name="role" value=**。
- 在终端应用中，前往 **EAP_HOME/bin** 目录。
- 创建一个具有 **admin**、**rest-all** 和 **kie-server** 角色的用户，用来登录到 Business Central。



注意

具有 **admin** 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 Server (KIE 服务器) REST 功能。对于需要访问 Business Central 中的 **Manage** 和 **Track** 视图的用户，需要有 **kie-server** 角色。

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 **admin** 的用户。

密码必须至少有 8 个字符，且必须至少包含一个数字，以及一个非字母数字字符，但不能包括 & (ampersand)。

4. 创建一个具有 **kie-server** 角色的用户，用于登录到 KIE Server。

```

$ ./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，您可以创建具有这两个角色的单个用户：

```

$ ./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的文件，并运行以下命令，其中 **<USER_DATA>.cli** 是包含用户数据的文件：

```

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

```

<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 服务器以连接到 BUSINESS CENTRAL



警告

本节提供了可用于测试目的的示例设置。其中一些值不适用于生产环境，并被标记为此类值。

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，或者需要在 Red Hat Process Automation Manager 环境中需要额外的 KIE 服务器，您必须配置 KIE 服务器来连接到 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](#) 以将其配置为连接到 Business Central。

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，角色为 **rest-all** 的用户
 - 在 KIE 服务器上，具有角色 **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，进入 **standalone-full.xml** 文件。例如，如果您为 Red Hat Process Automation Manager 使用 Red Hat JBoss EAP 安装，请转到 **\$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。

属性	值	备注
org.kie.server.controller.user	controllerUser	具有角色 rest-all 的用户名，可以登录到 Business Central。
org.kie.server.controller.pwd	controllerUser1234;	可以登录到 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 Server API 的 URL。

表 14.2. Business Central 实例的 JVM 属性

属性	值	备注
org.kie.server.user	controllerUser	带有角色 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;"/>
```

- 要验证 KIE 服务器是否已成功启动，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 服务器中运行红帽流程自动化管理器的更多信息，请参阅 [运行 Red Hat Process Automation Manager](#)。成功验证后，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. 验证注册是否成功：
 - a. 登录 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

为确保使用最佳线程数量，请将线程系统属性的值设置为 CPU 数加上一 的总和。

1. 在 Red Hat Process Automation Manager 安装目录中，进入 **standalone-full.xml** 文件。例如，如果您为 Red Hat Process Automation Manager 使用 Red Hat JBoss EAP 安装，请转到 **\$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 数加上一：

```

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。您可能需要根据额外的测试进行进一步微调。

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。您应该加密控制器和 KIE 服务器密码。如果 Business Central 和 KIE 服务器部署到不同的应用服务器，则两个应用服务器将使用密钥存储。

为您的密钥存储使用 Java CryptographyExtension KeyStore (JCEKS)，因为它支持对称密钥。使用 KeyTool (JDK 安装的一部分)来创建新的 JCEKS。



注意

如果没有使用 JCEKS 配置 KIE 服务器，KIE 服务器密码会以纯文本形式存储在系统属性中。

先决条件

- KIE 服务器安装在红帽 JBoss EAP 中。
- 安装了 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 Server 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>**：密钥存储将存储的路径
 - **<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>**：进程条目的别名
3. 出现提示时，输入您创建的 KIE Server 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，并替换下表中列出的占位符：

```
<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_ALIASES>"/>
<property name="kie.keystore.key.server.pwd" value="<KEY_SERVER_PWD>"/>
<property name="kie.keystore.key.ctrl.alias" value="<KEY_CONTROL_ALIASES>"/>
<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
kie.keystore.keyStoreURL	<KEYSTORE_URL>	要使用的 JCEKS 的 URL，例如 file:///home/kie/keystores/keystore.jceks
kie.keystore.keyStorePwd	<KEYSTORE_PWD>	JCEKS 的密码
kie.keystore.key.server.alias	<KEY_SERVER_ALIASES>	存储密码的 REST 服务的密钥别名
kie.keystore.key.server.pwd	<KEY_SERVER_PWD>	带有存储密码的 REST 服务别名的密码
kie.keystore.key.ctrl.alias	<KEY_CONTROL_ALIASES>	存储密码的默认 REST Process Automation Controller 的密钥别名
kie.keystore.key.ctrl.pwd	<KEY_CONTROL_PWD>	带有存储密码的默认 REST Process Automation Controller 的别名密码

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，DSA 加密算法由 Business Central 提供。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

完成以下任务之一来启用此系统属性：

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

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

- 启动 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 Server 的情况下部署了 Business Central，则您可以使用 **standalone.sh** 脚本启动 Business Central，而无需指定 **standalone-full.xml** 文件。在这种情况下，请确保在启动 Business Central 前对 **standalone.xml** 文件进行任何配置更改。

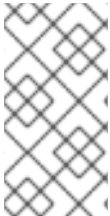
2. 在 Web 浏览器中，输入 **localhost:8080/business-central**。
 - 如果您将 Red Hat Process Automation Manager 配置为从域名运行，请将 **localhost** 替换为域名，例如：
http://www.example.com:8080/business-central
 - 如果您将 Red Hat Process Automation Manager 配置为在集群中运行，请将 **localhost** 替换为特定节点的 IP 地址，例如：
http://<node_IP_address>:8080/business-central
3. 输入您在安装过程中创建的 **admin** 用户凭证。此时会显示 Business Central 主页。
4. 选择 **Menu → Deploy → Execution Servers**。
5. 确认 **default-kieserver** 列在 **Server Configuration** 下。
6. 选择 **Menu → Design → Projects**。
7. 打开 **MySpace** 空间。
8. 点 **Try Samples → Mortgages → OK**。此时会显示 **Assets** 窗口。
9. 点 **Add Asset → Data Object**。
10. 在 **Data Object** 字段中输入 **MyDataObject**，然后点 **OK**。
11. 点 **Spaces → MySpace → Mortgages**，并确认 **MyDataObject** 是否在资产列表中。
12. 删除 **Mortgages** 项目。

第 18 章 运行 RED HAT PROCESS AUTOMATION MANAGER

使用这个流程在 Red Hat JBoss EAP 以独立模式运行 Red Hat Process Automation Manager。

先决条件

- 安装和配置 Red Hat Process Automation Manager。



注意

如果您更改了默认主机(**localhost**)或默认端口(**9990**), 在运行 Red Hat Process Automation Manager 之前, 您必须编辑 **business-central.war/WEB-INF/classes/datasource-management.properties** 和 **business-central.war/WEB-INF/classes/security-management.properties** 文件, 如 [Solution 3519551](#) 中所述。

流程

1. 在终端应用中, 前往 **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 Server 的情况下部署了 Business Central, 则您可以使用 **standalone.sh** 脚本启动 Business Central, 而无需指定 **standalone-full.xml** 文件。在这种情况下, 请确保在启动 Business Central 前对 **standalone.xml** 文件进行任何配置更改。

在 Linux 或基于 UNIX 的系统中 :

```
┌ $ ./standalone.sh
```

在 Windows 中 :

```
┌ standalone.bat
```

3. 在 Web 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, 请将 **localhost** 替换为域名, 例如 :

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

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

第 19 章 安装并运行无头进程 AUTOMATION MANAGER 控制器

您可以将 KIE 服务器配置为以受管或非受管模式运行。如果 KIE Server 是非受管服务器，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，流程自动化管理器控制器管理 KIE 服务器配置，并与 Process Automation Manager 控制器交互，以创建和维护 KIE 容器。

Business Central 有一个嵌入式流程 Automation Manager 控制器。如果您安装 Business Central，请使用 **Execution Server** 页面创建和维护 KIE 容器。如果要在不使用 Business Central 的情况下自动执行 KIE 服务器管理，您可以使用无头进程 Automation Manager 控制器。

19.1. 安装无头进程 AUTOMATION MANAGER 控制器

您可以安装无头进程 Automation Manager 控制器，并使用 REST API 或 KIE Server Java 客户端 API 来与之交互。

先决条件

- 提供了一个受支持的 Red Hat JBoss EAP 安装版本 7.4。红帽 JBoss EAP 安装的基础目录称为 **EAP_HOME**。
- 授予足够的用户权限来完成安装。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
2. 下载 Red Hat Process Automation Manager 7.12.0 Add Ons(**rhpm-7.12.0-add-ons.zip** 文件)。
3. 提取 **rhpm-7.12.0-add-ons.zip** 文件。**rhpm-7.12.0-controller-ee7.zip** 文件位于提取的目录中。
4. 将 **rhpm-7.12.0-controller-ee7.zip** 存档提取到临时目录。在以下示例中，此目录名为 **TEMP_DIR**。
5. 将 **TEMP_DIR/rhpm-7.12.0-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 Automation Manager 控制器部署的名称不会与 Red Hat JBoss EAP 实例中的现有部署冲突。

6. 将 **TEMP_DIR/rhpm-7.12.0-controller-ee7/SecurityPolicy/** 目录的内容复制到 **EAP_HOME/bin**。

- 当提示覆盖文件时，请选择 **是**。
- 在 **EAP_HOME/standalone/deployments/** 目录中，创建一个名为 **controller.war.dodeploy** 的空文件。此文件确保服务器启动时会自动部署无头进程 Automation Manager 控制器。

19.1.1. 创建无头进程 Automation Manager 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 Automation Manager 控制器安装在 Red Hat JBoss EAP 安装(**EAP_HOME**)的基础目录中。

流程

- 在终端应用中，前往 **EAP_HOME/bin** 目录。
- 输入以下命令，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 **admin** 的用户。

密码必须至少有 8 个字符，且必须至少包含一个数字，以及一个非字母数字字符，但不能包括 & (ampersand)。

- 记录您的用户名和密码。

19.1.2. 配置 KIE 服务器和无头进程 Automation Manager 控制器

如果 KIE 服务器由无头进程 Automation Manager 控制器管理，您必须在 KIE Server 安装中编辑 **standalone-full.xml** 文件，并在无头进程 Automation Manager 控制器安装中使用 **standalone.xml** 文件。

先决条件

- KIE 服务器安装在 **EAP_HOME** 中。
- 无头进程 Automation Manager 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 服务器和无头流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 Automation Manager 控制器，例如在开发环境中，请在共享 **standalone-full.xml** 文件中进行这些更改。

- 在 KIE Server 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>` 替换为用户的凭证，使用 **kie-server** 角色：

```
<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，使用 **kie-server** 角色。
 - 将 `<KIE_SERVER_ID>` 替换为 KIE 服务器安装的 ID 或名称，如 **rhpm-7.12.0-kie-server-1**。
 - 将 `<HOST>` 替换为 KIE Server 主机的 ID 或名称，如 **localhost** 或 **192.7.8.9**。
 - 将 `<PORT>` 替换为 KIE Server 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 服务器在启动时连接到此 URL。

19.2. 运行无头进程 AUTOMATION MANAGER 控制器

在 Red Hat JBoss EAP 上安装无头处理 Automation Manager 控制器后，使用此流程运行无头进程 Automation Manager 控制器。

先决条件

- 无头进程 Automation Manager 控制器在 Red Hat JBoss EAP 安装(**EAP_HOME**)的基础目录中安装和配置。

流程

1. 在终端应用中，前往 **EAP_HOME/bin**。
2. 如果您在与安装 KIE Server 的 Red Hat JBoss EAP 实例相同的 Red Hat JBoss EAP 实例上安装了无头处理 Automation Manager 控制器，请输入以下命令之一：

- 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
```

3. 如果您在与安装 KIE 服务器的 Red Hat JBoss EAP 实例独立的 Red Hat JBoss EAP 实例上安装了无头处理 Automation Manager 控制器，请使用 **standalone.sh** 脚本启动无头进程 Automation Manager 控制器：



注意

在这种情况下，请确保您对 **standalone.xml** 文件进行了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 Automation Manager 控制器是否在 Red Hat JBoss EAP 上工作，请输入以下命令，其中 **<CONTROLLER>** 和 **<CONTROLLER_PWD>** 是您在 [第 19.1.1 节“创建无头进程 Automation Manager 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>'
```

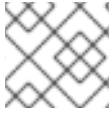


注意

另外，您可以使用 KIE Server Java API 客户端来访问无头进程 Automation Manager 控制器。

第 20 章 配置和运行独立 BUSINESS CENTRAL

您可以使用 Business Central 独立 JAR 文件来运行 Business Central，而不必将它部署到应用服务器。您可以使用示例配置文件来开箱即用运行 Business Central standalone JAR 文件，也可以根据您的要求自定义 sampfiles。



注意

此 JAR 文件仅在 Red Hat Enterprise Linux 上运行时被支持。

先决条件

- Red Hat Process Automation Manager 7.12.0 Business Central Standalone(**rhpam-7.12.0-business-central-standalone.jar**)和 Red Hat Process Automation Manager 7.12.0 Add Ons (**rhpam-7.12.0-add-ons.zip**)文件已从 Red Hat Process Automation Manager 7.12 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 **rhpam-7.12.0-addons.zip** 提取到临时目录。此存档包含 **rhpam-7.12.0-standalone-sample-configuration.zip** 文件。
2. 将 **rhpam-7.12.0-standalone-sample-configuration.zip** 文件提取到包含 **rhpam-7.12.0-business-central-standalone.jar** 文件的目录。**rhpam-7.12.0-standalone-sample-configuration.zip** 文件包含以下示例配置文件：
 - **application-script.cli**：用于添加用户和 kie 服务器系统属性的 Sample 脚本
 - **kie-fs-realm-users**：示例用户数据
您可以使用配置文件中提供的示例数据运行 **rhpam-7.12.0-business-central-standalone.jar** 文件，也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-7.12.0-business-central-standalone.jar --cli-script=application-script.cli
```

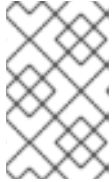
5. 要在运行 JAR 文件时设置应用程序属性，请在命令中包含 **-D<PROPERTY>=<VALUE>** 参数，其中 **<PROPERTY>** 是支持的应用程序属性的名称，**<VALUE>** 是属性值：

```
java -jar rhpam-7.12.0-business-central-standalone.jar --cli-script=application-script.cli -
D<PROPERTY>=<VALUE> -D<PROPERTY>=<VALUE>
```

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

```
java -jar rhpam-7.12.0-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，请将 **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 设置和存储库

当您创建红帽流程自动化管理器项目时，Business Central 将使用为 Business Central 配置的 Maven 存储库。您可以使用 Maven 全局或用户设置来指示所有 Red Hat Process Automation Manager 项目通过修改 Maven 项目对象模型(POM)文件(**pom.xml**)从公共红帽进程自动化管理器存储库检索依赖项。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 镜像供离线使用。

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 存储库的正确传输依赖项版本。

有关 Red Hat Business 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>7.12.0.redhat-00008</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，定义了面向用户的产品依赖项的版本，因此您不需要指定这些 `<dependency>` 元素的 `<version>` 的子元素。但是，您必须使用 `<dependency>` 元素来声明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关系。在您在 Business Central 中编写的项目中，业务中心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，请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，请声明以下依赖项：

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，请声明以下依赖项：

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

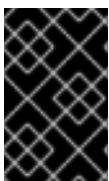
```
<dependency>
  <groupId>org.uberfire</groupId>
  <artifactId>uberfire-rest-client</artifactId>
</dependency>
```

- 在创建包含资产（如规则和进程定义）的 JAR 文件时，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 `<project>` 元素下的 **kjar** 打包类型。在以下示例中，`#{kie.version}` 是 [Red Hat Process Automation Manager](#) 和 [Maven 库版本之间的映射](#) 中列出的 Maven 库版本？

```
<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>
```

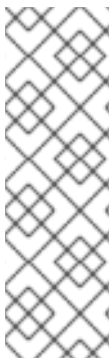
21.2. 为 BUSINESS CENTRAL 和 KIE 服务器配置外部 MAVEN 存储库

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 服务器访问和下载外部 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 需要工件不可变。因此，缺少已知安全漏洞补丁的工件会保留在存储库中，以避免依赖它们的构建。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

有关在 Red Hat OpenShift Container Platform 上为编写环境配置外部 Maven 存储库的详情，请查看以下文档：

- [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

- 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. 使用连接和访问外部存储库的详情，创建一个 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，进入 **standalone-full.xml** 文件。例如，如果您为 Red Hat Process Automation Manager 使用 Red Hat JBoss EAP 安装，请转到 **\$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. 启动或重新启动 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 **pom.xml** 文件中添加存储库信息。具体步骤请参阅 [打包和部署红帽流程自动化管理器项目](#)。

21.3. 准备 MAVEN 镜像存储库以供离线使用

如果您的 Red Hat Process Automation Manager 部署没有对公共互联网的传出访问，则必须使用所有必要的工件镜像准备 Maven 存储库，并将此存储库提供给您的环境。



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

- 具有一个可以传出访问公共互联网的计算机。

流程

1. 在与公共互联网的传出连接的计算机上，完成以下步骤：
2. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - **产品**：Red Hat Process Automation Manager
 - **Version**: 7.12
 - a. 下载并提取 **Red Hat Process Automation Manager 7.12.0 Offliner 内容列表(rhpam-7.12.0-offliner.zip)** 产品提供可交付的文件。
 - b. 将 **rhpam-7.12.0-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
```

此命令会创建 **存储库** 子目录，并将所需的工件下载到此子目录中。这是镜像存储库。

如果消息报告了一些下载失败，请再次运行相同的命令。如果下载再次失败，请联系红帽支持。

3. 如果您在 Business Central 之外开发服务，并且有额外的依赖项，请将依赖项添加到镜像存储库。如果将服务作为 Maven 项目开发，您可以使用以下步骤自动准备这些依赖项。在与公共互联网传出连接的计算机上完成步骤。
 - 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** 命令构建项目源。
 - c. 对于每个项目，输入以下命令来确保 Maven 下载项目生成的所有工件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` 替换为项目的 `pom.xml` 文件的路径。

- d. 将本地 Maven 缓存目录(`~/m2/repository`)的内容复制到所创建的 **存储库** 子目录中。
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 的目录中。此目录成为离线 Maven 镜像存储库。
 5. 为您的 Red Hat Process Automation Manager 部署创建和配置 **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，请添加缺少的标签。
 - 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>
```

将 `/path/to/repo` 替换为本地 Maven 镜像存储库目录的完整路径。

- 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>
```

将 `/path/to/repo` 替换为本地 Maven 镜像存储库目录的完整路径。

7. 将 Business Central 的 `kie.maven.offline.force` 属性设置为 `true`。有关为 Business Central 设置属性的说明，[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 是一个分布式版本控制系统。它将修订版本作为提交对象实施。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 来存储项目数据，包括规则和流程等资产。当您在 Business Central 中创建项目时，它将添加到嵌入在 Business Central 中的 Git 存储库中。如果您在其他 Git 存储库中拥有项目，则可以通过 Business Central 空格将这些项目导入到 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。
- 您有对该外部 Git 存储库的读取访问权限所需的凭证。

流程

1. 在 Business Central 中，点 **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**。
 - 从下拉列表中选择 **Import Project**。
4. 在 **Import Project** 窗口中，输入 Git 存储库的 URL 和凭证，该存储库包含您要导入的项目，然后单击 **Import**。项目添加到 Business Central Git 存储库中，并可从当前空间获得。

第 23 章 自定义 BUSINESS CENTRAL 的品牌

您可以通过将镜像替换为您自己的镜像来自定义 Business Central 登录页面和应用程序标题的品牌。

23.1. 自定义 BUSINESS CENTRAL 登录页面

您可以在 Business Central 登录页面上自定义公司徽标和项目徽标。

流程

1. 启动红帽 JBoss EAP 并在 Web 浏览器中打开 Business Central。
2. 将 SVG 格式镜像复制到红帽流程自动化管理器安装中的 **`EAP_HOME/standalone/deployments/ business-central.war/img/`** 目录中。
3. 在 **`EAP_HOME/standalone/deployments/ business-central.war/img/`** 目录中，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** 字段上方的项目徽标，请将默认镜像 **`BC_Logo.png`** 替换为一个新的 SVG 文件。
6. 强制完全重新加载登录页面，绕过缓存来查看更改。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 Red Hat JBoss EAP，在 Web 浏览器中打开 Business Central，并使用您的用户凭证登录。
2. 将 SVG 格式的新应用程序标头镜像复制到红帽流程自动化管理器 **安装** 中 **`EAP_HOME/standalone/deployments/ business-central.war/banner/`** 目录。
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`** 替换为新镜像的文件名：**`admin-and-config/`**

```

```

5. 强制完全重新加载登录页面，绕过缓存来查看更改。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，您可以通过红帽单点登录集成 LDAP 和 SSL。如需更多信息，请参阅 [Red Hat Single Sign-On Server Administration Guide](#)。

附录 A. BUSINESS CENTRAL 系统属性

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

Git 目录

使用以下属性来设置 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**: 启用或禁用 Git ketch。
- **org.uberfire.nio.git.hooks**: Git hook 目录的位置。

Git over HTTP

使用以下属性来配置通过 HTTP 访问 Git 存储库：

- **org.uberfire.nio.git.proxy.ssh.over.http**: 指定 SSH 是否应该使用 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（整数值）。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**: 启用或禁用 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，它将此属性用作主机标识符。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** : 如果启用了 HTTP 守护进程，它将此属性用作主机名标识符。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，它将此属性用作端口号。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

使用以下属性配置通过 HTTPS 访问 Git 存储库：

- **org.uberfire.nio.git.proxy.ssh.over.https**: 指定 SSH 是否使用 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** : 用户目录的位置。
- **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** : 如果启用了 HTTPS 守护进程，它将此属性用作主机标识符。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** : 如果启用了 HTTPS 守护进程，它将此属性用作主机名标识符。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** : 如果启用了 HTTPS 守护进程，它将此属性用作端口号。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**: 启用或禁用删除 eldest JGit iterations。
- **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 驱除阈值时间单位。

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**: 启用或禁用 Git 守护进程。默认值：**true**。
- **org.uberfire.nio.git.daemon.host** : 如果启用了 Git 守护进程，它将此属性用作本地主机标识符。默认值：**localhost**。
- **org.uberfire.nio.git.daemon.hostname** : 如果启用了 Git 守护进程，它将此属性用作本地主机名标识符。默认值：**localhost**
- **org.uberfire.nio.git.daemon.port** : 如果启用了 Git 守护进程，它将此属性用作端口号。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

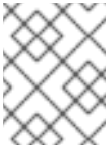
如果默认或分配的端口已经在使用中，则会自动选择一个新端口。确保端口可用，并检查日志以了解更多信息。

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, 它将此属性用作本地主机标识符。默认值: **localhost**。
- **org.uberfire.nio.git.ssh.hostname** : 如果启用了 SSH 守护进程, 它将此属性用作本地主机名标识符。默认值: **localhost**。
- **org.uberfire.nio.git.ssh.port** : 如果启用了 SSH 守护进程, 它将此属性用作端口号。默认值: **8001**。



注意

如果默认或分配的端口已经在使用中, 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 空闲超时。
- **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。示例: **git@github.com:user/repository.git**。
- **org.uberfire.nio.git.ssh.algorithm**: SSH 使用的算法。默认值: **RSA**。
- **org.uberfire.nio.git.gc.limit**: 设置 GC 限制。
- **org.uberfire.nio.git.ssh.ciphers** : 以逗号分隔的密码字符串。可用的密码有 **aes128-ctr,aes192-ctr,aes256-ctr,arcfour128,arcfour256,aes192-cbc,aes256-cbc**。如果没有使用属性, 则会加载所有可用的密码。
- **org.uberfire.nio.git.ssh.macs** : 以逗号分隔的消息身份验证代码(MAC)字符串。可用的 MAC 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 DSA 以外的 RSA 或任何算法, 请确保将应用服务器设置为使用 Bouncy Castle JCE 库。

KIE Server 节点和流程自动化管理器控制器

使用以下属性, 从 Process Automation Manager 控制器配置与 KIE 服务器节点的连接:

- **org.kie.server.controller** : 该 URL 用于连接到进程 Automation Manager 控制器。例如, **ws://localhost:8080/business-central/websocket/controller**。
- **org.kie.server.user** : 用于从 Process Automation Manager 控制器连接到 KIE Server 节点的用户名。只有在将此 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** : 用于从进程 Automation Manager 控制器连接到 KIE Server 节点的密码。只有在将此 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 和各种

使用以下属性配置 Maven 和其他各种功能:

- **kie.maven.offline.force**: 强制 Maven 的行为与离线一样。如果为 true，则禁用在线依赖项解析。默认值：**false**。



注意

仅将此属性用于 Business Central。如果您与任何其他组件共享运行时环境，请隔离配置并将其仅应用到 Business Central。

- **org.uberfire.gzip.enable** : 启用或禁用 **GzipFilter** 压缩过滤器上的 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**。前缀 **FULL_** 设置配置文件，并从管理员首选项隐藏配置集首选项。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。它指示 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：指向内部 **m2** 存储库的文件路径。
- **appformer.ssh.keystore** : 通过指定类名称来定义要与 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** : 使用默认的 SSH 密钥存储时，此属性定义用户的 SSH 公钥的存储文件夹。如果属性不可用，则密钥存储在 Business Central **.security** 文件夹中。
- **appformer.experimental.features** : 启用实验性功能框架。默认值：**false**。
- **org.kie.demo** : 从 GitHub 启用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** : 用于将 LDAP 主体名称映射到应用角色名称的 Regex 模式。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，因为应用角色名称会替换变量角色。
- **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**: 由密码加密使用的 Crypto 算法。默认值：**PBEWithMD5AndDES**。
- **org.uberfire.domain**: uberfire 使用的安全域名。默认值：**ApplicationRealm**。
- **org.guvnor.m2repo.dir**: 存储 Maven 存储库文件夹的 Place。默认值：**<working-directory>/repositories/kie**。
- **org.guvnor.project.gav.check.disabled** : 禁用组 ID、工件 ID 和版本(GAV)检查。默认值：**false**。
- **org.kie.build.disable-project-explorer** : 禁用在 Project Explorer 中自动构建所选项目。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为每个页面自定义资产数量。默认值 : 15。
- **org.kie.verification.disable-dtable-realtime-verification** : 禁用表的实时验证和验证。默认值 : **false**。

处理 Automation Manager 控制器

使用以下属性来配置如何连接到进程 Automation Manager 控制器 :

- **org.kie.workbench.controller** : 用于连接进程自动化管理器控制器的 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** : 用于连接进程 Automation Manager 控制器的令牌字符串。

Java 加密扩展 KeyStore (JCEKS)

使用以下属性来配置 JCEKS :

- **kie.keystore.keyStoreURL** : 用于加载 Java 加密扩展 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 控制器的别名密码。

渲染

使用以下属性在 Business Central 和 KIE Server rendered 表单间切换 :

- **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, 该表单渲染由 Business Central 执行。默认值 : **false**。
- **org.jbpm.wb.forms.renderer.name** : 使您能够在 Business Central 和 KIE Server rendered 表单之间进行切换。默认值 : **工作台**。

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 Red Hat Single Sign-On 集成](#)
- [进程服务入门](#)

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，您可以创建一个红帽流程自动化管理器集群环境，为开发和运行时环境提供高可用性和负载平衡。

先决条件

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

第 27 章 RED HAT PROCESS AUTOMATION MANAGER 集群

通过集群两个或多个计算机，您可以获得高可用性、增强的协作和负载平衡的优势。当单个计算机出现故障时，高可用性降低了数据丢失的几率。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数据的副本来填补差距。当失败的计算机再次上线时，它会恢复它在集群中的位置。

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

- [第 28 章 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 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。在集群环境中，如果开发人员操作某个节点并且该节点失败，则该开发人员的工作将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，至少包含一个 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. 在要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 配置红帽 JBoss EAP 7.4。
Red Hat Data Grid 从 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、搜索和分析大量数据，以近乎实时的时间。在 Red Hat Process Automation Manager 集群环境中，它可让您跨集群节点执行复杂和高效的搜索。
 - 配置 AMQ Broker，即 Java 消息传递服务器(JMS)代理。
JMS 代理是一个软件组件，它接收消息、将其存储在本地，并将消息转发到接收者。AMQ Broker 使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、企业版和 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 7.12，然后在每个要作为集群节点的系统中安装它们。
3. 在集群的每个节点上配置和启动 Business Central。

28.1. 安装和配置 RED HAT DATA GRID

要获得更有效的跨集群节点搜索，请安装 Red Hat Data Grid 并为 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 JBoss EAP 的 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. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：Data Grid

- **Version:** 8.1
2. 将 **Red Hat Data Grid 8.1.0 Server**(**redhat-datagrid-8.1.0-server.zip**)安装文件提取到系统上的首选位置。在以下示例中，提取的目录称为 **JDG_HOME**。
 3. 将 Red Hat Data Grid 更新至最新版本。如需更多信息，请参阅 Red Hat Data Grid [Red Hat Data Grid User Guide](#)。
 4. 要添加 Red Hat Data Grid 用户，请导航到 **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，请打开 **JDG_HOME/bin/server.conf** 文件并找到以下行：

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

6. 使用以下内容替换这一行：

```
-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. 使用以下内容替换这一行：

```
<hotrod-connector name="hotrod">
  <authentication>
    <saslm 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，请导航到 **JDG_HOME** 并输入以下命令：

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

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

28.2. 下载和配置 AMQ BROKER

Red Hat AMQ Broker 使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、企业版和 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. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：AMQ Broker
 - Version: 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. 将目录更改为 `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，请导航到 `amq-broker-7.8.2-bin/amq-broker-7.8.2/bin` 目录，然后输入以下命令：

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

28.3. 配置 NFS 版本 4 服务器

Business Central 集群环境需要具有共享文件系统的 NFS 版本 4 服务器，每个客户端节点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 共享的说明，请参阅 [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. 在服务器上，在文本编辑器中打开 `/etc/exports` 文件。
3. 将以下行添加到 `/etc/exports` 文件中，其中 `<HOST_LIST>` 是被授权连接到服务器的主机的 IP 地址和选项的列表：

```
/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)
```

这会创建一个带有 `rw,sync,no_root_squash` 最小选项的 `/opt/kie/data` 共享，这些选项是 NFS 所需的。



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，配置运行 Business Central 的所有节点时，必须使用其他名称。

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 7.12。

流程

1. 在集群的每个节点上下载 Red Hat JBoss EAP 7.4 :
 - a. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：企业应用平台
 - Version: 7.4
 - b. 点 Red Hat JBoss Enterprise Application Platform 7.4.1 旁边的 **Download**。（**JBEAP-7.4.1/jboss-eap-7.4.1.zip**）。
2. 提取 **jboss-eap-7.4.1.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. 在集群的每个节点上下载 Red Hat Process Automation Manager :
 - a. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面，然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
 - b. 下载 Red Hat Process Automation Manager 7.12.0 Business Central Deployable for Red Hat JBoss EAP 7 (**rhpm-7.12.0- business-central-central-eap7-deployable.zip**)。
5. 将 **rhpm-7.12.0- business-central-eap7-deployable.zip** 文件提取到临时目录中。在以下命令中，此目录名为 **TEMP_DIR**。
6. 将 **TEMP_DIR/rhpm-7.12.0- business-central-eap7-deployable/jboss-eap-7.4** 的内容复制到 **EAP_HOME**。
7. 下载并应用最新的红帽流程自动化管理器补丁（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 服务器管理指南 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。在集群的每个节点上完成这些步骤。



注意

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



注意

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

商业中部实例无法与每个 KIE 服务器的状态保持同步。例如，如果 KIE 服务器已启动但未同步，则 Business Central 将无法通过该实例部署。

先决条件

- 如 [第 28.1 节“安装和配置 Red Hat Data Grid”](#) 所述安装 Red Hat Data Grid 8.1。
- AMQ Broker 已安装并配置，如 [第 28.2 节“下载和配置 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。
- 提供一个共享文件夹的 NFS 服务器，如 [第 28.3 节“配置 NFS 版本 4 服务器”](#) 所述。

流程

1. 要将 NFS 共享的目录挂载为 **/data**，请以 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>** 替换为您要配置的共享名称（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>** 元素下的属性，并替换以下占位符：

- **<AMQ_USER & gt;** 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，进入 `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. 测试内部基础架构的高可用性(HA)

当您为 Business Central 创建生产就绪的高可用性(HA)时，您必须确保它满足可行 HA 环境的最低硬件和性能要求。HA 内部基础架构由以下四项主要组件组成：Business Central、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 个节点的网络环境被配置为以下布局：
 - 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：

- 在每个服务器节点的命令终端中，安装 **iPerf3**：

```
$ dnf install iperf3
```

- 在 NFS 服务器节点（服务器节点 3）的命令终端中，以服务器模式启动 **iPerf3**：

```
$ iperf3 -s
```

- 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.

- 将每个服务器节点的结果与以下最小值示例进行比较：

```
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
[ 5] 0.00-10.04  sec  677 MBytes  566 Mbits/sec
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. 验证磁盘是否为 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** 以验证磁盘是否为 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
```

28.7. 验证 RED HAT PROCESS AUTOMATION MANAGER 集群

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 Web 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。此时会显示 Business Central 主页。
3. 选择 **Menu → Design → Projects**。
4. 打开 **MySpace** 空间。
5. 点 **Try Samples → Mortgages → OK**。此时会显示 **Assets** 窗口。
6. 点 **Add Asset → Data Object**。
7. 在 **Data Object** 字段中输入 **MyDataObject**，然后点 **OK**。
8. 点 **Spaces → MySpace → Mortgages**，并确认 **MyDataObject** 是否在资产列表中。
9. 在网页浏览器中输入以下 URL，其中 `<node_IP_address>` 是集群的不同节点的地址：
`http://<node_IP_address>:8080/business-central`
10. 输入您用来登录到第一个节点上的 Business Central 的相同凭证，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** 是否在资产列表中。
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商业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平衡。如果在集群的一个节点中增加活动，则该活动可以在集群的其他节点之间共享，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 Server。然后，您可以为集群配置 Red Hat JBoss EAP 7.4，启动集群，并在每个集群节点上安装 KIE 服务器。

然后，您可以把无头进程 Automation Manager 控制器和智能路由器集群集群。



注意

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

商业中部实例无法与每个 KIE 服务器的状态保持同步。例如，如果 KIE 服务器已启动但未同步，则 Business Central 将无法通过该实例部署。

29.1. 下载并提取 RED HAT JBOSS EAP 7.4 和 KIE 服务器

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

流程

1. 在集群的每个节点上下载 Red Hat JBoss EAP 7.4 :
 - a. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：企业应用平台
 - Version: 7.4
 - b. 点 Red Hat JBoss Enterprise Application Platform 7.4.1 旁边的 **Download**。（**jboss-eap-7.4.1.zip**）。
2. 提取 **jboss-eap-7.4.1.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. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面，然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
 - b. 下载 适用于所有支持的 EE8 容器的 Red Hat Process Automation Manager 7.12.0 KIE Server (**rhcam-7.12.0-kie-server-ee8.zip**)。
5. 将 **rhcam-7.12.0-kie-server-ee8.zip** 归档提取到临时目录中。在以下示例中，此目录名为 **TEMP_DIR**。

- 将 **TEMP_DIR/rhpam-7.12.0-kie-kie-server-ee8/rhpam-7.12.0-kie-server-ee8/kie-server.war** 目录复制到 **EAP_HOME/standalone/deployments/**。



警告

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

- 将 **TEMP_DIR/rhpam-7.12.0-kie-kie-server-ee8/rhpam-7.12.0-kie-server-ee8/SecurityPolicy/** 复制到 **EAP_HOME/bin**。
- 当提示覆盖文件时，点 **替换**。
- 在 **EAP_HOME/standalone/deployments/** 目录中，创建一个名为 **kie-server.war.dodeploy** 的空文件。此文件确保服务器启动时自动部署 KIE 服务器。
- 下载并应用最新的红帽流程自动化管理器补丁（如果可用）。
- 前往 **EAP_HOME/bin** 目录。
- 创建一个具有 **kie-server** 角色的用户，用于登录到 KIE Server。

```

$ ./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'])"

```

- 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，然后启动集群。

流程

- 在作为此集群一部分的所有红帽 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。
- 在文本编辑器中打开 **EAP_HOME/standalone/configuration/standalone-full.xml** 文件。
- 编辑 **data-stores** 属性以及上方的 **timer-service thread-pool-name** :
 - datasource-jndi-name** 是此流程开头指定的数据库的 JNDI 名称。
 - 您可以为 **partition** 属性的值输入任何名称。但是，节点只能看到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。这样做可提高性能。例如，除了具有 100 个节点的集群外，所有 100 个节点都试图执行并刷新同一计时器，您可以通过为每个包括 5 个节点的组指定一个不同的分区名称来创建 20 个每个包括 5 个节点的集群。

- 将 **default-data-store** 属性值替换为 **ejb_timer_ds**。
- 以毫秒为单位设置 **refresh-interval** 的值，以指定 EJB 计时器连接到数据库的频率，以同步和加载要处理的任务。

```
<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和对应的 **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>** 是该数据库的凭证。

- 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

您必须为 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>` 是数据库的休眠位置。
支持以下 dialects：
 - DB2： `org.hibernate.dialect.DB2Dialect`
 - MSSQL： `org.hibernate.dialect.SQLServer2012Dialect`
 - 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.PostgresPlusDialect`

```

<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 AUTOMATION MANAGER 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，如果您没有安装 Business Central，您可以安装无头进程 Automation Manager 控制器，并使用 REST API 或 KIE Server Java 客户端 API 与之交互。

先决条件

- 提供了备份的 Red Hat 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**。
- 授予足够的用户权限来完成安装。
- 提供一个共享文件夹的 NFS 服务器，如第 28.3 节“配置 NFS 版本 4 服务器”所述。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
2. 下载 Red Hat Process Automation Manager 7.12.0 Add Ons(`rhpmam-7.12.0-add-ons.zip` 文件)。
3. 提取 `rhpmam-7.12.0-add-ons.zip` 文件。`rhpmam-7.12.0-controller-ee7.zip` 文件位于提取的目录中。
4. 将 `rhpmam-7.12.0-controller-ee7.zip` 存档提取到临时目录。在以下示例中，此目录名为 **TEMP_DIR**。
5. 将 `TEMP_DIR/rhpmam-7.12.0-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 Automation Manager 控制器部署的名称不会与 Red Hat JBoss EAP 实例中的现有部署冲突。

6. 将 `TEMP_DIR/rhpmam-7.12.0-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，创建一个名为 `controller.war.dodeploy` 的空文件。此文件确保服务器启动时会自动部署无头进程 Automation Manager 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>
```

模板文件包含特定部署场景的默认配置。

如果 `org.kie.server.controller.templatefile.watcher.enabled` 属性的值设置为 `true`，则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。如果此属性的值设置为 `false`，则只有在服务器重启时，才会检测到模板文件的更改。

11. 要启动无头进程 Automation Manager 控制器，请导航到 `EAP_HOME/bin` 并输入以下命令：

- 在 Linux 或基于 UNIX 的系统中：

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

- 在 Windows 中：

```
standalone.bat
```

29.4. 在 KIE 服务器节点上为 **RUNNING** 状态配置故障切换

默认情况下，当多个 KIE Server 节点在集群中运行时，如果一个 KIE Server 节点失败，其他可用的 KIE 服务器节点会从失败的 KIE Server 节点获取 **QUEUED** 或 **RETRYING** 状态的任何作业。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，以重新排队到另一个 KIE 服务器节点。如果失败的 KIE 服务器节点上的一个进程中有一个计时器启动节点，则进程只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** 的缓存的容器条目，其中包含一个名为 **nodes** 的缓存以及一个名为 **jobs** 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，并将以下依赖项添加到文件中：

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

- b. 下载 JAR 文件。

- i. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
- ii. 下载 Red Hat Process Automation Manager 7.12.0 Maven Repository(**rhpm-7.12.0-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/kie/server/kie-server-services-jbpm-cluster/7.59.0.Final-redhat-00006/kie-server-services-jbpm-cluster-7.59.0.Final-redhat-00006.jar** 文件复制到 **kie-server.war/WEB-INF/lib** 目录。

集群现在配置为使用 **RUNNING** 状态作业故障转移。

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，您可以在多个 KIE 服务器、客户端应用程序和其他组件之间用作集成层。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

当客户端应用程序请求并聚合结果时，从所有 KIE 服务器实例（每个组中一个实例）收集数据。

路由

作为单一端点，可以从客户端应用程序接收调用到任何服务，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。必须使用标准负载均衡工具在外部管理智能路由器集群的负载均衡请求。

身份验证

使用系统属性标志验证 KIE 服务器实例，并启用 HTTPS 流量。

环境管理

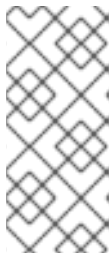
管理更改环境，如添加或删除服务器实例。

30.1. 使用智能路由器实现负载均衡 KIE 服务器实例

您可以使用智能路由器来聚合多个独立的 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执行。

先决条件

- 已安装多个 KIE 服务器实例。



注意

您不需要将 KIE 服务器配置为非托管智能路由器。

非受管 KIE 服务器实例不连接到控制器。例如，如果您将非受管 KIE 服务器实例连接到智能路由器，并将智能路由器注册到控制器，那么 Business Central 将使用智能路由器联系非受管 KIE 服务器实例。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
2. 下载 Red Hat Process Automation Manager 7.12.0 附加组件。
3. 将下载的 `rhpam-7.12.0-add-ons.zip` 文件提取到临时目录中。`rhpam-7.12.0-smart-router.jar` 文件位于提取的 `rhpam-7.12.0-add-ons` 目录中。
4. 将 `rhpam-7.12.0-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 rhpam-7.12.0-smart-router.jar
```

以上命令中的属性有以下默认值：

```
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，以启用 watcher 服务系统属性。



注意

您可以使用配置文件，而不是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 服务器实例中，将 **org.kie.server.router** 系统属性设置为智能路由器 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登录凭据。

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填充它，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>

```

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 SMART ROUTER

您可以为传输层安全(TLS)配置智能路由器(KIE 服务器路由器)支持来允许 HTTPS 流量。另外，您可以禁用到智能路由器的非安全 HTTP 连接。

先决条件

- KIE 服务器安装在红帽 JBoss EAP 7.4 集群的每个节点上。
- 安装并配置智能路由器。更多信息请参阅 [第 30.1 节“使用智能路由器实现负载均衡 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：

- 要启动启用了 TLS 支持和允许 HTTP 连接的 Smart Router，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-7.12.0-smart-router.jar
```

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。
- **<KEYSTORE_ALIAS >**：用于存储证书的别名名称。
- **<HTTPS_PORT >**：HTTPS 端口。默认 HTTPS 端口为 **9443**。
- 要启动启用了 TLS 支持和禁用 HTTP 连接的 Smart Router，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-7.12.0-smart-router.jar
```

当 **org.kie.server.router.port** 系统属性设置为 **0** 时，不会注册 HTTP 侦听器。如果配置了 TLS，并且 HTTP 侦听器没有注册，则 Smart Router 仅侦听 HTTPS 端口。



注意

如果没有配置 TLS，并且您通过将 **org.kie.server.router.port** 设置为 **0** 来禁用 HTTP，则会出现错误，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 Server Router)以进行端点身份验证。

先决条件

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

流程

- 要使用启用端点身份验证来启动 Smart Router，请配置管理凭证：
 - a. 在您的 KIE 服务器配置中添加以下属性：

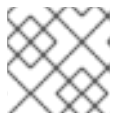
```
`org.kie.server.router.management.username`
`org.kie.server.router.management.password`
```

默认 **用户名** 是 KIE 服务器 ID。

- b. 在 Smart Router 配置中添加以下属性：

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

password 属性值为 **true** 或 **false**（默认）。



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被身份验证。

1. 可选：将用户添加到智能路由器。例如：

```
java -jar rhpam-7.12.0-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-7.12.0-smart-router.jar -removeUser <USERNAME>
```

30.4. 配置智能路由器行为

在有多个 KIE Servers 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 服务器的主机都使用“round-robin”方法发送请求。在以下示例环境中，每个 KIE 服务器都使用相同的 KJAR 部署，但每个 KJAR 版本不同：

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)**。

Smart Router 有三个组件，您可以修改以更改此行为：

ContainerResolver

负责在与服务器交互时使用的容器 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** 工具来实现这些组件：

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**，使智能路由器在所有可用 KIE 服务器间搜索 KJAR 进程的最新版本，并始终从该进程开始。这种情境意味着每个 KIE 服务器托管了一个 KJAR，每个版本将共享相同的别名。

由于 Smart Router 是可执行的 jar，因此要包含扩展，因此您需要修改该命令。例如：

```
java -cp LOCATION/router-ext-7.12.0.redhat-00008.jar:rhpm-7.12.0-smart-router.jar
org.kie.server.router.KieServerRouter
```

启动该服务后，您将看到日志输出，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

您可以使用配置文件，而不是在命令行中配置智能路由器设置。在这种情况下，设置（包括任何密码）在命令行终端和服务日志中不可见。

流程

1. 创建配置文件。此文件可以包含 **property=value** 格式的任意数量的行。该文件可以包含以下任何一个属性：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，以识别其他组件。	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.key store	TLS 连接的密钥存储文件	N/A
org.kie.server.router.tls.key store.password	TLS 连接的密钥存储密码	N/A
org.kie.server.router.tls.key store.keyalias	在密钥存储中引用 TLS 证书的别名名称	N/A
org.kie.server.router.repo	用于存储当前存储库的目录	当前工作目录
org.kie.router.identity.provider	用于使用智能路由器验证 KIE 服务器实例的可选自定义提供程序类。对于此接口的源代码，此类必须实施 org.kie.server.router.identity.IdentityManager 接口，请参阅 GitHub 存储库 。	N/A
org.kie.server.controller	连接到控制器的 URL	N/A

属性名称	描述	默认值
org.kie.server.controller.user	连接到控制器的用户名	kieserver
org.kie.server.controller.pwd	连接到控制器的密码	kieserver1!
org.kie.server.controller.token	用于连接到控制器的身份验证令牌	N/A
org.kie.server.controller.retry.interval	如果失败，重试与控制器的连接的时间间隔（以秒为单位）	10
org.kie.server.controller.retry.limit	与控制器连接的最大重试次数（如果失败）	无限
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 ，则智能路由器需要密码来验证来自 KIE Server 的连接	false

2. 使用以下命令启动智能路由器：

```
java -Dorg.kie.server.router.config.file=<CONFIG_FILE> -jar rhpam-7.12.0-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。

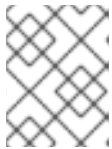
在应用服务器上配置数据库之前，您必须为 Quartz 准备数据库，以创建 Quartz 表，该表将存放计时器数据和 Quartz 定义文件。

先决条件

- 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. 在数据库中创建 Quartz 表，以使用 DDL 脚本为您的数据库同步计时器事件。DDL 脚本位于 **QUARTZ_HOME/docs/dbTables** 的提取补充 ZIP 存档中。



注意

创建前，包含单词 **drop** 的脚本，如 **quartz_tables_drop_db2.sql** 丢弃 Quartz 表。

2. 在 **JBOSS_HOME/MODE/configuration/** 目录中创建 Quartz 配置文件 **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.PostgreSQLDelegate
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/quartzNotManagedDS
```

```
# 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，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 **org.quartz.properties** 属性中 **quartz-definition.properties** 文件的绝对路径。
4. 可选：要为 Quartz 触发器配置重试和延迟数量，请更新以下系统属性：
 - **org.jbpm.timer.quartz.retries**（默认值为 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，以参与流程引擎的事务。
- 非受管数据源查找要触发的计时器，而无需任何事务处理

Red Hat Process Automation Manager 业务应用程序假定 Quartz 数据库(schema)将与 Red Hat Process Automation Manager 表共存，因此生成用于 Quartz 事务操作的数据源。

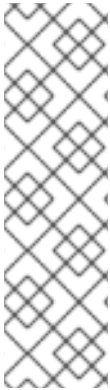
必须配置其他（非事务）数据源，但它应指向与主数据源相同的数据库。

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 7.12。



注意

对 Red Hat JBoss Web Server 和 Apache Tomcat 上的 Red Hat Process Automation Manager 的支持现在处于维护阶段。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。
- 红帽只会发布具有严重影响和关键任务程序错误修复的合格安全补丁。

在未来，红帽可能会要求客户迁移到与红帽混合云策略兼容的新平台和产品组件。

先决条件

- 您已查看了 [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 服务器的详情，请参考 [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。它旨在使业务和 IT 用户能够更有效地协作，从而轻松、快速地更改业务应用程序。

Red Hat Process Automation Manager 由 Business Central 和 KIE 服务器组成。

- KIE Server 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计划问题。KIE 服务器为业务资产提供运行时环境，并访问资产存储库中存储的数据（知识存储）。
- Business Central 是您创建和管理 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。商业中心也作为独立 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 的中度和大型应用程序而设计。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 Red Hat JBoss Web Server 安装中，您可以安装 KIE Server 和无头流程自动化管理器控制器。或者，您可以运行独立的 Business Central JAR 文件。

本文档中的说明解释了如何在 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 Server 上安装和配置 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

Red Hat Process Automation Manager 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) 中安装 Business Central。商业中心也作为独立 JAR 文件提供。您可以使用 Business Central 独立 JAR 文件来运行 Business Central，而不必将它部署到应用服务器。
- **KIE Server** 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计划问题。您可以在 Red Hat JBoss EAP 实例、Red Hat JBoss EAP 集群中、在 OpenShift 上安装 KIE 服务器，在一个 Oracle WebLogic 服务器实例、IBM WebSphere Application Server 实例中，或作为 Spring Boot 应用程序的一部分。您可以将 KIE 服务器配置为以受管或非受管模式运行。如果 KIE Server 是非受管服务器，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，流程自动化管理器控制器管理 KIE 服务器配置，并与 Process Automation Manager 控制器交互，以创建和维护 KIE 容器。

在 Red Hat JBoss Web Server 安装中，您可以安装 KIE Server 和无头流程自动化管理器控制器。或者，您可以运行独立的 Business Central JAR 文件。

33.2. RED HAT PROCESS AUTOMATION MANAGER 角色和用户

要访问 Business Central 或 KIE Server，您必须创建用户并在服务器启动前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 服务器都在单一实例上运行，则对 Business Central 进行身份验证的用户也可以访问 KIE 服务器。

但是，如果 Business Central 和 KIE 服务器在不同实例上运行，则对 Business Central 进行身份验证的用户必须单独通过身份验证才能访问 KIE 服务器。例如，如果一个在 Business Central 上通过身份验证但未通过 KIE Server 进行身份验证的用户尝试查看或管理 Business Central 中的进程定义，则会在日志文件中记录 401 错误，以及从远程服务器加载数据的 **Invalid 凭证**。请联系您的系统管理员。信息出现在 Business Central 中。

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



注意

admin, analyst, developer, manager, process-admin, user, 和 **rest-all** 角色为 Business Central 保留。**kie-server** 角色为 KIE 服务器保留。因此，可用的角色可能会因 Business Central、KIE Server 还是安装这两者而异。

- **admin**：具有管理 **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。它们具有对应用程序所需的更改的完整访问权限。具有 **admin** 角色的用户可以访问 Red Hat Process Automation Manager 中的所有区域。
- **分析师**：具有 **分析者** 角色的用户有权访问所有高级别功能。他们可以对他们的项目进行建模和执行。但是，这些用户无法在 **Design → Projects** 视图中为空格或删除空格添加贡献者。带有 **analyst** 角色的用户没有访问 **Deploy → Execution Servers** 视图（面向管理员）的访问权限。但是，当这些用户访问 Library 透视图时，可以使用 **Deploy** 按钮。
- **开发人员**：具有 **developer** 角色的用户几乎可以访问所有的功能，并且可以管理规则、模型、流程、表单和仪表盘。他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）才会在具有 **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。这些用户通常对业务流程及其性能、业务指示器和其他业务相关报告相关的统计信息感兴趣。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。他们可完全访问业务流程、业务任务和执行错误。这些用户还可以查看业务报告，并可访问 Task Inbox 列表。
- **用户** : 具有用户角色 **的用户** 可以处理 Task Inbox 列表，其中包含属于当前运行的进程的商业任务。具有此角色的用户可以查看进程和任务报告和管理进程。
- **rest-all** : 拥有 **rest-all** 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 Server REST 功能。对于需要访问 Business Central 中的 **Manage** 和 **Track** 视图的用户，这个角色是必须的。

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 Server 部署 ZIP 文件。如果您要运行 Business Central 而不将其部署到应用服务器，请下载 Business Central Standalone JAR 文件。

下载满足您的环境和安装要求的 Red Hat Process Automation Manager 发行版本。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
2. 根据您首选的安装方法下载以下产品发行版之一：



注意

您只需要下载其中一个发行版。

- 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 7.12.0 Installer(rhpam-installer-7.12.0.jar)**。安装程序图形用户界面指导您完成安装过程。
- 要使用可部署的 ZIP 文件在 Red Hat JBoss Web Server 上安装 KIE 服务器，请下载以下文件：
 - Red Hat Process Automation Manager 7.12.0 Add Ons(**rhpam-7.12.0-add-ons.zip**)
 - Red Hat Process Automation Manager 7.12.0 Maven Repository(**rhpam-7.12.0-maven-repository.zip**)
- 要运行 Business Central，无需将它部署到应用程序服务器，请下载 **Red Hat Process Automation Manager 7.12.0 Business Central Standalone (rhpam-7.12.0-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：

按照以下部分之一的说明操作：

- [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。您可以使用它来在现有 Red Hat JBoss Web Server 5.5.1 服务器安装中安装 Red Hat Process Automation Manager。

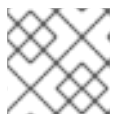


注意

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 7.12.0 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](#)。
- 提供了受支持的 Red Hat JBoss Web Server 5.5.1 服务器安装。
- 授予足够的用户权限来完成安装。



注意

确保您已使用对 Tomcat 具有写权限的用户登录。

- 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 旨在与 UTF-8 编码一起工作。如果底层 JVM 使用了不同的编码系统，则可能会出现意外的错误。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，请参阅 [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 文件的目录，然后输入以下命令：

■

```
java -jar rhpam-installer-7.12.0.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据。要防止此要求，请在安装命令中添加 **izpack.mode=privileged** 选项：

```
java -Dizpack.mode=privileged -jar
rhpam-installer-7.12.0.jar
```

另外，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-7.12.0.jar
```

图形安装程序显示启动屏幕和许可证协议页面。

2. 单击 **I accept the terms of this license agreement**，然后单击 **Next**。
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，然后点 **Next**。
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 Server 和无头流程 Automation Manager 控制器。无头进程 Automation Manager 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，请安装无头进程的 Automation Manager 控制器。
5. 创建用户，然后单击下一步。默认情况下，如果您在同一容器中安装 Business Central 和 KIE Server，则新用户将获得 **admin**、**kie-server** 和 **rest-all** 角色。如果您只安装 KIE Server，则会给出 **kie-server** 角色。访问 KIE 服务器 REST 功能需要 **kie-server** 角色。



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 **admin** 的用户。

密码必须至少有 8 个字符，且必须至少包含一个数字，以及一个非字母数字字符，但不能包括 & (ampersand)。

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

6. 在 **Installation Overview** 页面中，查看您要安装的组件并点 **Next** 开始安装。
7. 安装完成后，单击 **下一步**。
8. 如果安装了 KIE Server，则 **组件安装** 下会显示 **Configure Runtime** 步骤。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 **Perform advanced configuration**，您可以选择配置数据库设置或自定义某些 KIE Server 选项。

9. 如果您选择了 **Customize database settings**, 在 **JDBC Drive Configuration** 页面中指定数据源 JDBC 驱动程序供应商, 请选择一个或多个驱动程序 JAR 文件, 然后单击 **Next**。
数据源是一种对象, 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, 以确保服务器与指定数据库之间的正确数据交换。
10. 如果您选择了 **Customize KIE Server properties**, 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 Server 属性的名称。
 - 取消选择您要禁用的任何 KIE 服务器功能。
11. 点 **Next** 配置运行时环境。
12. 当在屏幕的顶部出现 **Processing finished** 时, 点 **Next** 来完成安装。
13. 可选 : 点 **Generate Installation Script and Properties File** 将安装数据保存在 XML 文件中, 然后点 **Done**。
安装程序生成两个文件。**auto.xml** 文件自动执行将来的安装, **auto.xml.variables** 文件存储用户密码和其他敏感变量。使用 **auto.xml** 文件在多个具有相同类型的服务器以及与原始安装相同的配置的情况下重复 Red Hat Process Automation Manager 安装。如有必要, 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-7.12.0.jar <path-to-auto.xml-file>
```

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



注意

如果使用 Microsoft SQL Server, 请确保已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。建议的配置是通过输入以下语句来打开

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 7.12.0 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](#)。

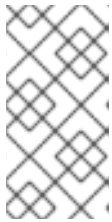
- 提供了受支持的 Red Hat JBoss Web Server 5.5.1 服务器安装。
- 授予足够的用户权限来完成安装。



注意

确保您已使用对 Tomcat 具有写权限的用户登录。

- 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 旨在与 UTF-8 编码一起工作。如果底层 JVM 使用了不同的编码系统，则可能会出现意外的错误。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，请参阅 [附录 B, Business Central 系统属性](#)。

流程

1. 在终端窗口中，进入您下载安装程序文件的目录，并输入以下命令：

```
java -jar rhpam-installer-7.12.0.jar -console
```

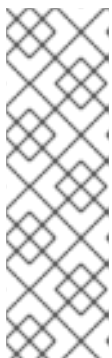
命令行交互流程将启动并显示最终用户许可协议。

```
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。
安装程序将在提供的位置验证安装位置。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

当您创建用户名和密码时，请确保指定的用户名与角色或组群的任何已知标题没有冲突。例如，如果有一个名为 **admin** 的角色，则不应创建一个名为 **admin** 的用户。

密码必须至少有 8 个字符，且必须至少包含一个数字，以及一个非字母数字字符（不能包括字符 **&**）。

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

5. 安装完成后，您将看到这个信息：

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

6. 输入 **y** 创建包含安装数据的 XML 文件，或输入 **n** 来完成安装。如果输入 **y**，系统会提示您指定 XML 文件的路径。
7. 输入路径或按 Enter 键接受推荐的路径。
安装程序生成两个文件。**auto.xml** 文件自动执行将来的安装，**auto.xml.variables** 文件存储用户密码和其他敏感变量。在多个系统上使用 **auto.xml** 文件，在与原始安装相同的配置相同的服务器上轻松重复 Red Hat Process Automation Manager 安装。如有必要，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-7.12.0.jar <path-to-auto.xml-file>
```

8. 如果您只安装 KIE Server，请重复这些步骤，以便在单独的服务器上安装无头进程 Automation Manager 控制器。



注意

如果使用 Microsoft SQL Server，请确保已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。建议的配置是通过输入以下语句来打开 ALLOW_SNAPSHOT_ISOLATION 和 READ_COMMITTED_SNAPSHOT：

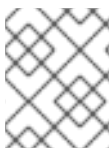
```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON  
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

第 36 章 KIE SERVER ZIP 文件安装和配置

您可以使用客户门户网站上的 `rhpm-7.12.0-kie-server-jws.zip` 文件安装 KIE 服务器，该文件位于 **Red Hat Process Automation Manager 7.12.0 Add Ons (rhpm-7.12.0-add-ons.zip)** 文件，然后在 Red Hat JBoss Web Server 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，并访问资产存储库中存储的数据（知识存储）。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 Server。



注意

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

- 下载以下文件，如 [第 34 章 下载 Red Hat Process Automation Manager 安装文件](#) 所述：
 - Red Hat Process Automation Manager 7.12.0 Add Ons(`rhpm-7.12.0-add-ons.zip`)
 - Red Hat Process Automation Manager 7.12.0 Maven Repository(`rhpm-7.12.0-maven-repository.zip`)
- 提供了受支持的 Red Hat JBoss Web Server 5.5.1 服务器安装。Red Hat JBoss Web Server 安装的基础目录称为 **JWS_HOME**。
- 授予足够的用户权限来完成安装。

流程

1. 提取 `rhpm-7.12.0-add-ons.zip` 文件。
2. 从提取的 `rhpm-7.12.0-add-ons.zip` 文件中提取以下文件：
 - `rhpm-7.12.0-kie-server-jws.zip`
 - `rhpm-7.12.0-process-engine.zip`

在以下说明中，包含提取的 `rhpm-7.12.0-kie-server-jws.zip` 文件的目录称为 **JWS_TEMP_DIR**，包含提取的 `rhpm-7.12.0-process-engine.zip` 文件的目录名为 **ENGINE_TEMP_DIR**。

3. 将 `JWS_TEMP_DIR/rhpm-7.12.0-kie-server-jws/kie-server.war` 目录复制到 `JWS_HOME/tomcat/webapps` 目录。



注意

确保您复制的 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` 目录中删除 `.war` 扩展。
5. 将 `kie-tomcat-integration-7.59.0.Final-redhat-00006.jar` 文件从 **ENGINE_TEMP_DIR** 目录移到 `JWS_HOME/tomcat/lib` 目录。

- 将 `jboss-jacc-api-<VERSION>.jar`, `slf4j-api-<VERSION>.jar`, 和 `slf4j-jdk14-<VERSION>.jar` 文件从 `ENGINE_TEMP_DIR/lib` 目录移动到 `JWS_HOME/tomcat/lib` 目录, 其中 `<VERSION>` 是 `lib` 目录中的版本工件文件名。
- 在上个 Valve 定义后, 将以下行添加到 `JWS_HOME/tomcat/conf/server.xml` 文件中的 `<host>` 元素中:

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

- 在文本编辑器中打开 `JWS_HOME/tomcat/conf/tomcat-users.xml` 文件。
- 将用户和角色添加到 `JWS_HOME/tomcat/conf/tomcat-users.xml` 文件中。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, 使用逗号分隔角色:

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

- 在 `JWS_HOME/tomcat/bin` 目录中完成以下步骤之一:

- 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"
```

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

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

注意

如果使用 Microsoft SQL Server, 请确保为数据库有适当的事务隔离。如果没有, 您可能会遇到死锁。建议的配置是通过输入以下语句来打开 `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 规格, 用于将 Java 编写的程序连接到数据库。数据源是一种对象, 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, 以确保服务器与指定数据库之间的正确数据交换。

通常, 使用红帽流程自动化管理器的解决方案在单个事务中管理多个资源。JMS 用于异步作业、事件和计时器, 例如: Red Hat Process Automation Manager 尽可能在数据源中有一个 XA 驱动程序, 以确保数据原子性和结果一致。如果不同架构的事务代码存在于监听器内, 或源自 jBPM 引擎提供的 hook, 则

还需要 XA 驱动程序。

除非您没有参与单个事务的多个资源，否则不要使用非 XA 数据源。



注意

对于生产环境，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 7.12.x Maven Repository (**rhpam-7.12.0-maven-repository.zip**)和 Red Hat Process Automation Manager 7.12.x Add-Ons (**rhpam-7.12.0-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 Hibernate dialects 之一：
 - DB2 : **org.hibernate.dialect.DB2Dialect**
 - MSSQL : **org.hibernate.dialect.SQLServer2012Dialect**
 - 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.PostgresPlusDialect**
 - archive: **org.hibernate.dialect.SybaseASE157Dialect**

流程

1. 完成以下步骤以准备数据库：
 - a. 将 **rhpam-7.12.0-add-ons.zip** 提取到临时目录中，如 **TEMP_DIR**。
 - b. 提取 **TEMP_DIR/rhpam-7.12.0-migration-tool.zip**。
 - c. 将您的当前目录更改为 **TEMP_DIR/rhpam-7.12.0-migration-tool/ddl-scripts** 目录。此目录包含多种数据库类型的 DDL 脚本。
 - d. 将数据库类型的 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

如果您将 PostgreSQL 或 Oracle 与 Spring Boot 结合使用，您必须导入相应的 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 **rhpm-7.12.0-maven-repository.zip** 离线 Maven 存储库文件。
3. 将提取的离线 Maven 存储库中的以下库复制到 **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** 文件中配置池 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。检查您的 JDBC 驱动程序的文档，以确定要设置的属性。

- a. 配置 XA 数据源，无需池功能。此 XA 数据源用于创建到目标数据库的新连接。在以下示例中，AXx 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
<datasource.dbName>	数据库的名称。
<datasource.port>	数据库的端口号。

变量	描述
<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 ，而其他数据库（如 H2 数据库）此属性为 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，该源依赖于 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.TransactionDataSourceFactory"
testOnBorrow="true"
transactionManager="TransactionManager"
transactionSynchronizationRegistry="TransactionSynchronizationRegistry"
type="javax.sql.XADataSource"
username="${datasource.username}"
password="${datasource.password}"
xaDataSource="xads"
/>
</Context>
```

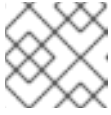
数据源现在包括在 **java:comp/env/poolingXaDs** JNDI 名称下，并通过 **org.kie.server.persistence.ds** 系统属性传递给 KIE Server，如后续步骤中所述。



注意

池数据源配置依赖于之前在 **kie-server** 应用程序中的 **context.xml** 文件中配置的其他资源，特别是 **TransactionManager** 和 **TransactionSynchronizationRegistry**。

6. 将 KIE 服务器配置为使用数据源：
- 在文本编辑器中打开以下脚本之一：

**注意**

setenv.sh 或 **setenv.bat** 脚本应该已经存在。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>** 是数据库的 Hibernate 空间：

```
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。您应该加密控制器和 KIE 服务器密码。如果 Business Central 和 KIE 服务器部署到不同的应用服务器，则两个应用服务器将使用密钥存储。

为您的密钥存储使用 Java CryptographyExtension KeyStore (JCEKS)，因为它支持对称密钥。使用 KeyTool (JDK 安装的一部分)来创建新的 JCEKS。



注意

如果没有使用 JCEKS 配置 KIE 服务器，KIE 服务器密码会以纯文本形式存储在系统属性中。

先决条件

- KIE 服务器在 Red Hat JBoss Web Server 中安装。
- 安装了 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. 将带有 **kie-server** 角色的 KIE Server 用户添加到 ***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_ALIASE> -storepass
<KEYSTORE_PASSWORD> -storetype JCEKS
```

在本例中，替换以下变量：

- **<KEYSTORE_PATH>**：密钥存储将存储的路径
 - **<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_ALIASE>**：进程条目的别名
4. 出现提示时，输入您创建的 KIE Server 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，并替换变量，如下表所述：



注意

如果在 Red Hat JBoss Web Server 的独立实例中安装 Business Central 或独立控制器，请不要将 **kie.keystore.key.server.alias** 和 **kie.keystore.key.server.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 上，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，例如 file:///home/kie/keystores/keystore.jceks
kie.keystore.keyStorePwd	<KEYSTORE_PWD>	JCEKS 的密码
kie.keystore.key.server.alias	<KEY_SERVER_ALIAS>	存储密码的 REST 服务的密钥别名
kie.keystore.key.server.pwd	<KEY_SERVER_PWD>	带有存储密码的 REST 服务别名的密码
kie.keystore.key.ctrl.alias	<KEY_CONTROL_ALIAS>	存储密码的默认 REST Process Automation Controller 的密钥别名
kie.keystore.key.ctrl.pwd	<KEY_CONTROL_PWD>	带有存储密码的默认 REST Process Automation Controller 的别名密码

- 启动 KIE 服务器以验证配置。

第 38 章 验证 KIE 服务器安装

验证 KIE Server 是否已正确安装。

先决条件

- KIE 服务器已安装并配置。

流程

1. 要启动 KIE 服务器，在 **JWS_HOME/tomcat/bin** 目录中输入以下命令之一：
 - 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，请在 Web 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 AUTOMATION MANAGER 控制器

您可以将 KIE 服务器配置为以受管或非受管模式运行。如果 KIE Server 是非受管服务器，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，流程自动化管理器控制器管理 KIE 服务器配置，并与 Process Automation Manager 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 **Execution Server** 页面创建和维护 KIE 容器。但是，如果您没有安装 Business Central，您可以安装无头进程 Automation Manager 控制器，并使用 REST API 或 KIE Server Java 客户端 API 与之交互。

先决条件

- Red Hat Process Automation Manager 7.12.0 Add Ons(**rhpm-7.12.0-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。
- 提供了 Red Hat JBoss Web Server 5.5.1 服务器安装。Red Hat JBoss Web Server 安装的基础目录称为 **JWS_HOME**。
- 授予足够的用户权限来完成安装。

流程

1. 提取 **rhpm-7.12.0-add-ons.zip** 文件。**rhpm-7.12.0-controller-jws.zip** 文件位于提取的目录中。
2. 将 **rhpm-7.12.0-controller-jws.zip** 归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-7.12.0-controller-jws.zip/controller.war** 目录复制到 **JWS_HOME/tomcat/webapps** 目录。



注意

确保您复制的 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** 目录中删除 **.war** 扩展。
5. 将 **TEMP_DIR/rhpm-7.12.0-controller-jws/SecurityPolicy/** 目录的内容复制到 **JWS_HOME/bin**
6. 当提示覆盖文件时，请选择 **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 上，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>** 替换为您在此流程中定义的 **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 Automation Manager 控制器的实例的 **JWS_HOME/tomcat/bin** 目录中，创建一个带有以下内容的 read **setenv.sh** 文件，其中 **<USERNAME>** 是 KIE Server 用户，**<USER_PWD>** 是该用户的密码：

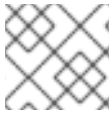

```
CATALINA_OPTS="-Dorg.kie.server.user=<USERNAME> -Dorg.kie.server.pwd=<USER_PWD>"
```
 10. 要启动无头进程 Automation Manager 控制器，在 **JWS_HOME/tomcat/bin** 目录中输入以下命令之一：
 - 在 Linux 或基于 UNIX 的系统中：


```
$. /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 Automation Manager 控制器是否正常工作，请在 web 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 配置和运行独立 BUSINESS CENTRAL

您可以使用 Business Central 独立 JAR 文件来运行 Business Central，而不必将它部署到应用服务器。您可以使用示例配置文件来开箱即用运行 Business Central standalone JAR 文件，也可以根据您的要求自定义 sampfiles。



注意

此 JAR 文件仅在 Red Hat Enterprise Linux 上运行时被支持。

先决条件

- Red Hat Process Automation Manager 7.12.0 Business Central Standalone(**rhpm-7.12.0-business-central-standalone.jar**)和 Red Hat Process Automation Manager 7.12.0 Add Ons (**rhpm-7.12.0-add-ons.zip**)文件已从 Red Hat Process Automation Manager 7.12 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-7.12.0-addons.zip** 提取到临时目录。此存档包含 **rhpm-7.12.0-standalone-sample-configuration.zip** 文件。
2. 将 **rhpm-7.12.0-standalone-sample-configuration.zip** 文件提取到包含 **rhpm-7.12.0-business-central-standalone.jar** 文件的目录。**rhpm-7.12.0-standalone-sample-configuration.zip** 文件包含以下示例配置文件：
 - **application-script.cli**：用于添加用户和 kie 服务器系统属性的 Sample 脚本
 - **kie-fs-realm-users**：示例用户数据
您可以使用配置文件中提供的示例数据运行 **rhpm-7.12.0-business-central-standalone.jar** 文件，也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 rhpm-7.12.0-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>** 是支持的应用程序属性的名称，**<VALUE>** 是属性值：

```
java -jar rhpm-7.12.0-business-central-standalone.jar --cli-script=application-script.cli -
D<PROPERTY>=<VALUE> -D<PROPERTY>=<VALUE>
```

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

```
java -jar rhpam-7.12.0-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，请将 **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 设置和存储库

当您创建红帽流程自动化管理器项目时，Business Central 将使用为 Business Central 配置的 Maven 存储库。您可以使用 Maven 全局或用户设置来指示所有 Red Hat Process Automation Manager 项目通过修改 Maven 项目对象模型(POM)文件(**pom.xml**)从公共红帽进程自动化管理器存储库检索依赖项。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 镜像供离线使用。

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(POM.XML)配置 MAVEN

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 项目](#)。

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。
在编译和打包项目时，Maven 从 Maven 存储库下载所有 JAR 文件和依赖 JAR 文件。

在 http://maven.apache.org/maven-v4_0_0.xsd 找到 **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 项目 [设置参考](#)。

流程

1. 在 **settings.xml** 文件中，声明 Red Hat Process Automation Manager 项目使用的存储库。通常，这是您从红帽客户门户网站下载的在线红帽流程自动化管理器 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存储库。
2. 确保 Business Central 或 KIE 服务器被配置为使用 **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 Red Hat JBoss Web 服务器上，对于 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。对于独立的 Business Central，请输入以下命令：

```
java -jar rhpam-7.12.0-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 存储库的正确传输依赖项版本。

有关 Red Hat Business 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>7.12.0.redhat-00008</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，定义了面向用户的产品依赖项的版本，因此您不需要指定这些 `<dependency>` 元素的 `<version>` 的子元素。但是，您必须使用 `<dependency>` 元素来声明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关系。在您在 Business Central 中编写的项目中，业务中心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，请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，请声明以下依赖项：

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，请声明以下依赖项：

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（如规则和进程定义）的 JAR 文件时，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 `<project>` 元素下的 **kjar** 打包类型。在以下示例中，`${kie.version}` 是 Red Hat Process Automation Manager 和 Maven 库版本之间的映射中列出的 Maven 库版本？

```
<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 镜像存储库以供离线使用

如果您的 Red Hat Process Automation Manager 部署没有对公共互联网的传出访问，则必须使用所有必要的工件镜像准备 Maven 存储库，并将此存储库提供给您的环境。



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

- 具有一个可以传出访问公共互联网的计算机。

流程

- 在与公共互联网的传出连接的计算机上，完成以下步骤：
- 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：Red Hat Process Automation Manager
 - Version: 7.12
 - 下载并提取 Red Hat Process Automation Manager 7.12.0 Offliner 内容列表(`rhpm-7.12.0-offliner.zip`)产品提供可交付的文件。
 - 将 `rhpm-7.12.0-offliner.zip` 文件的内容提取到任何目录中。
 - 进入该目录并输入以下命令：

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


此命令会创建 **存储库** 子目录，并将所需的工件下载到此子目录中。这是镜像存储库。

如果消息报告了一些下载失败，请再次运行相同的命令。如果下载再次失败，请联系红帽支持。

- 如果您在 Business Central 之外开发服务，并且有额外的依赖项，请将依赖项添加到镜像存储库。如果将服务作为 Maven 项目开发，您可以使用以下步骤自动准备这些依赖项。在与公共互联网传出连接的计算机上完成步骤。

- 创建本地 Maven 缓存目录(`~/.m2/repository`)的备份，然后清除该目录。
- 使用 `mvn clean install` 命令构建项目源。
- 对于每个项目，输入以下命令来确保 Maven 下载项目生成的所有工件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` 替换为项目的 `pom.xml` 文件的路径。

- 将本地 Maven 缓存目录(`~/.m2/repository`)的内容复制到所创建的 **存储库** 子目录中。
- 将 **存储库** 子目录的内容复制到您在其上部署 Red Hat Process Automation Manager 的目录中。此目录成为离线 Maven 镜像存储库。
 - 为您的 Red Hat Process Automation Manager 部署创建和配置 `settings.xml` 文件，如 [第 41.2 节“修改 Maven 设置文件”](#) 所述。
 - 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>
```

将 `/path/to/repo` 替换为本地 Maven 镜像存储库目录的完整路径。

- 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>
```

将 **/path/to/repo** 替换为本地 Maven 镜像存储库目录的完整路径。

第 42 章 从 GIT 存储库导入项目

Git 是一个分布式版本控制系统。它将修订版本作为提交对象实施。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 来存储项目数据，包括规则和流程等资产。当您在 Business Central 中创建项目时，它将添加到嵌入在 Business Central 中的 Git 存储库中。如果您在其他 Git 存储库中拥有项目，则可以通过 Business Central 空格将这些项目导入到 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。
- 您有对该外部 Git 存储库的读取访问权限所需的凭证。

流程

1. 在 Business Central 中，点 **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**。
 - 从下拉列表中选择 **Import Project**。
4. 在 **Import Project** 窗口中，输入 Git 存储库的 URL 和凭证，该存储库包含您要导入的项目，然后单击 **Import**。项目添加到 Business Central Git 存储库中，并可从当前空间获得。

第 43 章 集成 LDAP 和 SSL

通过 Red Hat Process Automation Manager，您可以通过红帽单点登录集成 LDAP 和 SSL。如需更多信息，请参阅 [Red Hat Single Sign-On Server Administration Guide](#)。

附录 B. BUSINESS CENTRAL 系统属性

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

Git 目录

使用以下属性来设置 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**: 启用或禁用 Git ketch。
- **org.uberfire.nio.git.hooks**: Git hook 目录的位置。

Git over HTTP

使用以下属性来配置通过 HTTP 访问 Git 存储库：

- **org.uberfire.nio.git.proxy.ssh.over.http**: 指定 SSH 是否应该使用 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（整数值）。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**: 启用或禁用 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，它将此属性用作主机标识符。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** : 如果启用了 HTTP 守护进程，它将此属性用作主机名标识符。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，它将此属性用作端口号。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

使用以下属性配置通过 HTTPS 访问 Git 存储库：

- **org.uberfire.nio.git.proxy.ssh.over.https**: 指定 SSH 是否使用 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** : 用户目录的位置。
- **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** : 如果启用了 HTTPS 守护进程，它将此属性用作主机标识符。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** : 如果启用了 HTTPS 守护进程，它将此属性用作主机名标识符。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** : 如果启用了 HTTPS 守护进程，它将此属性用作端口号。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**: 启用或禁用删除 eldest JGit iterations。
- **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 驱除阈值时间单位。

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**: 启用或禁用 Git 守护进程。默认值：**true**。
- **org.uberfire.nio.git.daemon.host** : 如果启用了 Git 守护进程，它将此属性用作本地主机标识符。默认值：**localhost**。
- **org.uberfire.nio.git.daemon.hostname** : 如果启用了 Git 守护进程，它将此属性用作本地主机名标识符。默认值：**localhost**
- **org.uberfire.nio.git.daemon.port** : 如果启用了 Git 守护进程，它将此属性用作端口号。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

如果默认或分配的端口已经在使用中，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, 它将此属性用作本地主机标识符。默认值: **localhost**。
- **org.uberfire.nio.git.ssh.hostname** : 如果启用了 SSH 守护进程, 它将此属性用作本地主机名标识符。默认值: **localhost**。
- **org.uberfire.nio.git.ssh.port** : 如果启用了 SSH 守护进程, 它将此属性用作端口号。默认值: **8001**。



注意

如果默认或分配的端口已经在使用中, 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 空闲超时。
- **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。示例: **git@github.com:user/repository.git**。
- **org.uberfire.nio.git.ssh.algorithm**: SSH 使用的算法。默认值: **RSA**。
- **org.uberfire.nio.git.gc.limit**: 设置 GC 限制。
- **org.uberfire.nio.git.ssh.ciphers** : 以逗号分隔的密码字符串。可用的密码有 **aes128-ctr,aes192-ctr,aes256-ctr,arcfour128,arcfour256,aes192-cbc,aes256-cbc**。如果没有使用属性, 则会加载所有可用的密码。
- **org.uberfire.nio.git.ssh.macs** : 以逗号分隔的消息身份验证代码(MAC)字符串。可用的 MAC 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 DSA 以外的 RSA 或任何算法, 请确保将应用服务器设置为使用 Bouncy Castle JCE 库。

KIE Server 节点和流程自动化管理器控制器

使用以下属性, 从 Process Automation Manager 控制器配置与 KIE 服务器节点的连接:

- **org.kie.server.controller** : 该 URL 用于连接到进程 Automation Manager 控制器。例如, **ws://localhost:8080/business-central/websocket/controller**。
- **org.kie.server.user** : 用于从 Process Automation Manager 控制器连接到 KIE Server 节点的用户名。只有在将此 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** : 用于从进程 Automation Manager 控制器连接到 KIE Server 节点的密码。只有在将此 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 和各种

使用以下属性配置 Maven 和其他各种功能:

- **kie.maven.offline.force**: 强制 Maven 的行为与离线一样。如果为 true，则禁用在线依赖项解析。默认值：**false**。



注意

仅将此属性用于 Business Central。如果您与任何其他组件共享运行时环境，请隔离配置并将其仅应用到 Business Central。

- **org.uberfire.gzip.enable** : 启用或禁用 **GzipFilter** 压缩过滤器上的 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**。前缀 **FULL_** 设置配置文件，并从管理员首选项隐藏配置集首选项。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。它指示 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：指向内部 **m2** 存储库的文件路径。
- **appformer.ssh.keystore** : 通过指定类名称来定义要与 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** : 使用默认的 SSH 密钥存储时，此属性定义用户的 SSH 公钥的存储文件夹。如果属性不可用，则密钥存储在 Business Central **.security** 文件夹中。
- **appformer.experimental.features** : 启用实验性功能框架。默认值：**false**。
- **org.kie.demo** : 从 GitHub 启用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** : 用于将 LDAP 主体名称映射到应用角色名称的 Regex 模式。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，因为应用角色名称会替换变量角色。
- **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**: 由密码加密使用的 Crypto 算法。默认值：**PBEWithMD5AndDES**。
- **org.uberfire.domain**: uberfire 使用的安全域名。默认值：**ApplicationRealm**。
- **org.guvnor.m2repo.dir**: 存储 Maven 存储库文件夹的 Place。默认值：**< ;working-directory>/repositories/kie**。
- **org.guvnor.project.gav.check.disabled** : 禁用组 ID、工件 ID 和版本(GAV)检查。默认值：**false**。
- **org.kie.build.disable-project-explorer** : 禁用在 Project Explorer 中自动构建所选项目。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为每个页面自定义资产数量。默认值 : 15。
- **org.kie.verification.disable-dtable-realtime-verification** : 禁用表的实时验证和验证。默认值 : **false**。

处理 Automation Manager 控制器

使用以下属性来配置如何连接到进程 Automation Manager 控制器 :

- **org.kie.workbench.controller** : 用于连接进程自动化管理器控制器的 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** : 用于连接进程 Automation Manager 控制器的令牌字符串。

Java 加密扩展 KeyStore (JCEKS)

使用以下属性来配置 JCEKS :

- **kie.keystore.keyStoreURL** : 用于加载 Java 加密扩展 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 控制器的别名密码。

渲染

使用以下属性在 Business Central 和 KIE Server rendered 表单间切换 :

- **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, 该表单渲染由 Business Central 执行。默认值 : **false**。
- **org.jbpm.wb.forms.renderer.name** : 使您能够在 Business Central 和 KIE Server rendered 表单之间进行切换。默认值 : **工作台**。

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 Red Hat KIE 服务器配置 IBM WebSphere Application Server，并在那个 IBM 服务器实例上安装 KIE Server。



注意

对 IBM WebSphere Application Server 上的 Red Hat Process Automation Manager 的支持现在处于维护阶段。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。
- 红帽只会发布具有严重影响和关键任务程序错误修复的合格安全补丁。

在未来，红帽可能会要求客户迁移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 9.0 或更高版本。有关完整的安装说明，请参阅 [IBM WebSphere Application Server 产品页面](#)。
- 您可以访问 WebSphere Integrated Solutions 控制台，通常在 **<http://<HOST>:9060/ibm/console>** 处访问。

第 44 章 KIE SERVER

KIE Server 是存储并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 REST、Java 消息服务(JMS)或 Java 客户端应用程序可用的接口来实例化和执行规则，以及通过解析程序管理流程、作业和红帽构建 OptaPlanner 功能。

创建为 Web 可部署的 WAR 文件，KIE 服务器可以部署到任何 Web 容器中。KIE 服务器的当前版本包括在 Red Hat Decision Manager 和 Red Hat Process Automation Manager 的默认扩展中。

KIE 服务器占用量较低，内存消耗较少，因此可以在云实例上轻松部署。此服务器的每个实例都可以打开和实例化多个容器，允许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 是一个灵活的安全 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，且完全与 Java EE 7 兼容。

第 46 章 安装并运行 IBM WEBSHERE APPLICATION SERVER

必须安装并运行 IBM WebSphere Application Server，才能应用容纳 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页面](#) 下载 [IBM 安装管理器](#) 1.8.5 或更高版本。安装 IBM WebSphere 需要 IBM 安装管理器。

2. 提取下载的存档，并以 root 用户身份在新目录中运行以下命令：

```
sudo ./install
```

IBM 安装管理器 将打开。

3. 进入 **File** → **Preferences**，点 **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 产品页面的在线产品存储库中找到所有存储库 URL](#)。
5. 在命令终端中，导航到安装期间指定的 IBM WebSphere Application Server 文件夹位置。
6. 更改到 **/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。该配置文件包含服务器在运行时环境中进程的所有文件，并可更改这些文件。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 **Servers** → **Server Types** → **IBM WebSphere Application Servers** 控制台中定义的 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，要为系统上运行的 IBM WebSphere 的本地实例启动 WebSphere Integrated Solutions 控制台，请在网页浏览器中输入以下 URL：

```
http://localhost:9060/ibm/console
```

9. 当显示 WebSphere Integrated Solutions Console 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 Application Server 部署 KIE 服务器之前，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 Application Server 已安装且正在运行。
- 已登陆到 WebSphere Integrated Solutions 控制台。

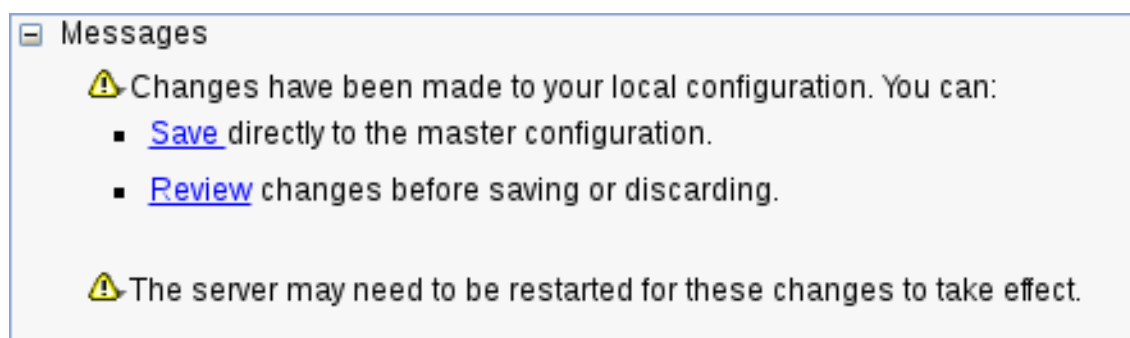
47.1. 启用管理安全性

您必须在 WebSphere Integrated Solutions 控制台中启用管理安全性，以便获得创建用户和组所需的权限。

流程

1. 在 WebSphere Integrated Solutions 控制台中，点 **Security** → **Global Security**，并确保选择了 **Enable Application Security** 选项。这可能会在服务器级别选择并覆盖。
2. 点 **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 存储库**。
4. 点 **Next**
5. 输入 **Primary administrative user name** 和 **Password**。
6. 单击 "下一步"，然后单击 "完成"。
7. 点 **Messages** 窗口中的 **Save**，将您的更改保存到主配置中。

图 47.1. 保存安全更新



8. 在命令终端中，导航到安装期间指定的 IBM WebSphere Application Server **/bin** 目录位置，并运行以下命令来停止并重启 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 `Servers` → `Server Types` → `IBM WebSphere Application Servers` 控制台中定义的 IBM WebSphere Application Server 名称。

47.2. 在 IBM WEBSHERE APPLICATION SERVER 中配置 JDBC 数据源

数据源是一种对象，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，以确保服务器与指定数据库之间的正确数据交换。

通常，使用红帽流程自动化管理器的解决方案在单个事务中管理多个资源。JMS 用于异步作业、事件和计时器，例如：Red Hat Process Automation Manager 尽可能在数据源中有一个 XA 驱动程序，以确保数据原子性和结果一致。如果不同架构的事务代码存在于监听器内，或源自 jBPM 引擎提供的 hook，则还需要 XA 驱动程序。

除非您没有参与单个事务的多个资源，否则不要使用非 XA 数据源。

先决条件

- 您要用于创建数据库连接的 JDBC 提供程序在您要在其上部署数据源的所有服务器上进行配置。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. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
2. 下载 [Red Hat Process Automation Manager 7.12.0 附加组件](#)。
3. 完成以下步骤以准备数据库：
 - a. 将 `rhpm-7.12.0-add-ons.zip` 提取到临时目录中，如 `TEMP_DIR`。
 - b. 提取 `TEMP_DIR/rhpm-7.12.0-migration-tool.zip`。
 - c. 将您的当前目录更改为 `TEMP_DIR/rhpm-7.12.0-migration-tool/ddl-scripts` 目录。此目录包含多种数据库类型的 DDL 脚本。
 - d. 将数据库类型的 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

如果您将 PostgreSQL 或 Oracle 与 Spring Boot 结合使用，您必须导入相应的 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 Integrated Solutions 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。您可以选择单元、节点、集群或服务器。
6. 点 **New** 打开 **Create a data source** 向导。
7. 在 **Data source name** 字段中输入唯一名称，并在 **JNDI name** 字段中输入 Java Naming and Directory Interface (JNDI) 名称。应用服务器使用 JNDI 名称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之间分配重复的 JNDI 名称，如数据源与 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. 单击 **Next**，再选择 **Select an existing JDBC provider if the provider is created, or select Create new JDBC provider** 来定义新提供程序的详细信息。（建议此前创建 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，如果容器管理持久性(CMP)企业级 Bean 必须访问此数据源，请单击 **Use this data source in container managed persistence (CMP)**，然后填写任何其他数据库特定属性。
10. 可选：为数据源配置安全别名。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：
 - **组件管理的身份验证别名**：指定在组件资源引用 res-auth 值是 **Application** 时使用的身份验证别名。要定义新别名，请导航到 **Related Items → J2EE Connector Architecture (J2C) 身份验证数据条目**。组件管理的别名代表应用程序中指定的 ID 和密码组合，用于数据源身份验证。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**：仅在组件资源引用上没有登录配置时使用。当 res-auth 值设置为 **Container** 时，指定登录配置和组件资源引用的相关属性是定义身份验证策略的首选方法。如果您指定了 **DefaultPrincipalMapping** 登录配置，则 associated 属性是一个 JAAS - J2C 身份验证数据条目别名。
 - **容器管理的身份验证别名**：仅在组件资源引用上没有登录配置时使用。当 res-auth 值设置为 **Container** 时，登录配置以及组件资源引用上的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，以及为同一范围配置的任何其他数据源。

有关 IBM WebSphere Application Server 数据源的更多信息，请参阅 IBM 知识库中配置 [JDBC 供应商和数据源](#)。

47.3. 配置 JAVA 消息服务(JMS)

Java 消息服务(JMS)是一种 Java API，KIE 服务器用于与其他应用服务器（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，并将 IBM WebSphere Application Server 添加为成员，才能使用 JMS。

流程

1. 在 WebSphere Integrated Solutions 控制台中，导航到 **Service Integration → Buses → New**。
2. 输入一个新的总线名称并清除 **Bus Security** 选项。

- 单击 **Next**，然后单击 **Finish** 以创建服务总线。
- 选择您创建的服务总线。
- 在 **Topology** 下，选择 **Bus Members** → **Add**。
- 在 **Add a New Bus Member** 向导中，选择 IBM WebSphere Application Server 以及用于持久性的消息类型。您还可以指定消息存储的属性。
- 单击 **Finish** 以添加新总线成员。

47.3.2. 创建 JMS 连接工厂

要启用 KIE Server 的消息传递，您必须创建特定的 JMS 连接工厂来发送和接收消息。

先决条件

- 您已为 IBM WebSphere Application Server 创建服务总线。

流程

- 在 WebSphere Integrated Solutions 控制台中，导航到 **Resources** → **JMS** → **Connection Factories**。
- 选择正确的范围，再单击 **New**。
- 选择 **Default Messaging Provider** 选项，再单击 **OK**。
- 对于以下每个所需的连接工厂，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。保留剩余选项的默认值。
- 单击 **Apply**，然后单击 **Save** 以将更改保存到主配置，然后对每个所需工厂重复此操作。

47.3.2.1. KIE 服务器的 JMS 连接工厂

下表列出了启用 KIE 服务器 JMS 消息传递所需的 Java 消息服务(JMS)连接工厂：

表 47.1. KIE 服务器所需的 JMS 连接工厂

名称	默认值	用于
KIE.SERVER.REQUEST	jms/cf/KIE.SERVER.REQUEST	将所有请求发送到 KIE 服务器
KIE.SERVER.RESPONSE	jms/cf/KIE.SERVER.RESPONSE	接收 KIE 服务器生成的所有响应
KIE.SERVER.EXECUTOR	jms/cf/KIE.SERVER.EXECUTOR	KIE Server executor 服务

47.3.3. 创建 JMS 队列

JMS 队列是点对点消息传递的目的地端点。您必须创建特定的 JMS 队列，才能使用 KIE 服务器启用 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 控制台中，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** 以将更改保存到主配置，然后对每个所需队列重复此操作。

47.3.3.1. KIE 服务器的 JMS 队列

下表列出了启用使用 KIE Server 的 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

名称	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 Server executor 服务

47.3.4. 创建 JMS 激活规格

需要 JMS 激活规范来桥接队列，以及启用 JMS 的消息驱动型 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 Integrated Solutions 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. 对于以下每个所需的激活规范，请输入激活规范的名称（如 **KIE.SERVER.REQUEST**）和 JNDI 名称（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**，再输入对应队列的名称作为目标 **查找**（例如，**jms/KIE.SERVER.REQUEST**）。
6. 单击 **Apply**，然后单击 **Save** 以将更改保存到主配置，然后对每个所需的激活规格重复此操作。

47.3.4.1. KIE 服务器的 JMS 激活规格

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)激活规范：

表 47.3. KIE 服务器所需的 JMS 激活规格

名称	默认值	用于
KIE.SERVER.REQUEST	jms/activation/KIE.SERVER.REQUEST	将所有请求发送到 KIE 服务器
KIE.SERVER.RESPONSE	jms/activation/KIE.SERVER.RESPONSE	接收 KIE 服务器生成的所有响应
KIE.SERVER.EXECUTOR	jms/activation/KIE.SERVER.EXECUTOR	KIE Server executor 服务

47.4. 在 IBM WEBSHERE APPLICATION SERVER 中设置系统属性

在部署 KIE 服务器前，设置本节中列出的系统属性。

流程

1. 在 WebSphere Integrated Solutions 控制台中，导航到 **Servers** → **Server Types** → **IBM WebSphere Application Servers**。
2. 在应用服务器列表中，选择您要在其上部署 KIE Server 的服务器。
3. 在 **Server Infrastructure** 下，点 **Java and Process Management** → **Process Definition**。

图 47.2. WebSphere 配置页面

Application servers > server1

Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.

Runtime Configuration

General Properties

Name: server1

Node name: dhcp-4-116Node01

Run in development mode

Parallel start

Start components as needed

Access to internal server classes: Allow

Server-specific Application Settings

ClassLoader policy: Multiple

Class loading mode: Classes loaded with parent class loader first

Apply OK Reset Cancel

Container Settings

- Session management
- SIP Container Settings
- Web Container Settings
- Portlet Container Settings
- EJB Container Settings
- Container Services
- Business Process Services

Applications

- Installed applications

Server messaging

- Messaging engines
- Messaging engine inbound transports
- WebSphere MQ link inbound transports
- SIB service

Server Infrastructure

- Java and Process Management
 - Class loader
 - Process definition**
 - Process execution
- Administration
 - Java SDKs

4. 在附加属性下，点 Java Virtual Machine。

图 47.3. 进程定义配置页面

这将打开用于启动 IBM WebSphere 的 JVM 的配置属性。

- 将 **Initial 堆大小** 和 **最大堆大小** 设置为 **2048**，然后单击 **Apply** 以增加 Java 虚拟机(JVM)内存大小。KIE 服务器已使用这些值进行测试。如果您没有增大 JVM 内存大小，IBM WebSphere Application Server 在部署 KIE 服务器时冻结或导致部署错误。
- 在 **附加属性** 下，点 **Custom Properties**。
- 单击 **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

名称	值	描述
kie.server.jms.queues.response	jms/KIE.SERVER.RESPONSE	用于 KIE 服务器使用的响应的 JMS 队列的 JNDI 名称。
org.kie.server.domain	WSLogin	JAAS LoginContext 域，用于在使用 JMS 时对用户进行身份验证。

名称	值	描述
<code>org.kie.server.persistence.ds</code>	<code>jdbc/jbpm</code>	KIE 服务器的数据源 JNDI 名称。
<code>org.kie.server.persistence.tm</code>	<code>org.hibernate.engine.transaction.jta.platform.internal.WebSphereJtaPlatform</code>	用于设置 Hibernate 属性的事务管理器平台。
<code>org.kie.server.persistence.dialect</code>	示例： <code>org.hibernate.dialect.H2Dialect</code>	指定要使用的 Hibernate dialect。根据数据源设置。
<code>org.kie.executor.jms.queue</code>	<code>JMS/KIE.SERVER.EXECUTOR</code>	KIE 服务器的作业执行程序 JMS 队列。
<code>org.kie.executor.jms.cf</code>	<code>jms/cf/KIE.SERVER.EXECUTOR</code>	KIE 服务器的作业执行者 JMS 连接工厂。
<code>org.kie.server.router</code>	示例： <code>http://localhost:9000</code>	(可选) 指定应用服务器是集群 KIE 服务器环境中的一个或多个 KIE 服务器路由器 (Smart routers) 的一个或多个 URL。
<code>org.jboss.logging.provider</code>	<code>jdk</code>	只有在环境中安装 CA SiteMinder TAI (SMTAI) 时，才需要此属性。使用此属性强制 Hibernate 使用 JDK 而不是 log4j 在 Dashbuilder 中记录。 CA SiteMinder TAI (SMTAI) 包含旧版本的 log4j ，这会导致冲突。

8. 点 **Save** 将更改保存到主配置。

47.5. 停止并重启 IBM WEBSHERE APPLICATION SERVER

在 IBM WebSphere Application Server 中配置了所有必要的系统属性后，停止并重启 IBM 服务器，以确保应用配置。

流程

在命令终端中，导航到安装期间指定的 IBM WebSphere Application Server `/bin` 目录位置，并运行以下命令来停止并重启 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 **Servers → Server Types → IBM WebSphere Application Servers** 控制台中定义的 IBM WebSphere Application Server 名称。

第 48 章 使用 IBM WEBSHERE APPLICATION SERVER 安装并运行 KIE SERVER

在 IBM WebSphere Application 服务器中配置了所有必要的系统属性后，您可以使用 IBM WebSphere 安装 KIE Server，以简化红帽流程自动化管理器应用程序管理。

先决条件

- 配置了 IBM WebSphere Application Server 实例，如 [第 47 章 为 KIE 服务器配置 IBM WebSphere Application Server](#) 所述。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
2. 下载 [适用于所有支持的 EE7 Containers 的 Red Hat Process Automation Manager 7.12.0 KIE Server](#)。
3. 将 `rhpm-7.12.0-kie-server-ee7.zip` 存档提取到临时目录。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-7.12.0-kie-server-ee7/kie-server.war` 目录。
 - b. 选择 `TEMP_DIR/rhpm-7.12.0-kie-server-ee7/kie-server.war` 目录的内容并创建 `kie-server.zip` 文件。
 - c. 将 `kie-server.zip` 重命名为 `kie-server.war`。这是您将用于部署 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其中部署的位置。
5. 在 WebSphere Integrated Solutions 控制台中，导航到 **Applications** → **Application Types** → **WebSphere Enterprise Applications**。
6. 单击 **InstCll**。
7. 导航到您重新打包的 `kie-server.war` 文件，再选择要上传的文件。
8. 选择 **Fast Path** 并点 **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，然后点 **Next**。
11. 对于 **Message-Driven Beans** 的 **Bind Listeners**，为这两个 Bean 选择 **Activation Specification**，在 **Target Resource JNDI Name** 字段中输入 `jms/activation/KIE.SERVER.REQUEST`，再输入 `jms/ct/KIE.SERVER.REQUEST` JNDI 名称作为 `KIE.SERVER.REQUEST` 连接。

12. 在 **Map Virtual Hosts for Web Modules** 部分中，保留默认值并单击 **Next**。
13. 将上下文 `root` 设置为 **kie-server**。
14. 在 **Modules** 的 **Metadata** 中，保留默认值并单击 **Next**。
15. 单击 **Finish** 以安装 KIE Server，然后单击 **Save** 以将更改保存到主配置。

48.1. 创建 KIE SERVER 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 Integrated Solutions 控制台中，点 **Users and Groups** → **Manage Groups**。
2. 在 **Manage Groups** 屏幕中，点 **Create**。
3. 在 **Create a Group** 屏幕中，在 **组名称** 框中输入 **kie-server**，然后单击 **Create**。
4. 要创建要添加到 **kie-server** 组的用户，请点击 **Users and Groups** → **Manage Users**。
5. 在 **Create a User** 屏幕中，完成所需信息。
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 **Create a User** 屏幕上，单击 **Create**。

48.2. 映射 KIE SERVER 组和角色

安装 KIE 服务器后，您必须将 **kie-server** 角色映射到 WebSphere Integrated Solutions 控制台中的 **kie-server** 组，以运行 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 **Details Properties** 下，点 **Security Role to User/Group Mapping**。
3. 选择 **kie-server** 角色，然后单击 **Map Groups** 以搜索 **kie-server** 组。
4. 将 **kie-server** 组从 **Available** 列表移到 **Selected** 列表中，再单击 **OK**。

此映射可让 IBM WebSphere Application Server **kie-server** 组中的用户访问 KIE Server。

5. 点 **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，将父类设置为最后一个加载。

流程

1. 导航到 **Applications → Application Types → WebSphere Enterprise Applications**，再点 **kie-server**。
2. 在左侧的 **Details Properties** 标题下，单击 **Class Loading** 和 **Update Detection**。
3. 在属性中，将 *Class Loader Order* 更改为 **使用本地类加载器的类，首先（父）** 和 *WAR Class Loader Policy* **用于应用程序**。
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 Server 组映射后，验证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 Automation Manager 控制器设置了所有必需的系统属性。
- 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：
 - 导航到 KIE Server URL **http://<HOST>:<PORT>/kie-server**。
 - 向 **http://<HOST>:<PORT>/kie-server/services/rest/server** 发送 **GET** 请求，以检查 KIE Server REST API 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 Server 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 安装并运行无头流程 AUTOMATION MANAGER 控制器

要使用 KIE Server REST API 或 Java 客户端 API 与 KIE 服务器交互，请使用 IBM WebSphere Application Server 安装无头流程 Automation Manager 控制器。无头进程 Automation Manager 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 Automation Manager 控制器创建和维护容器并执行其他服务器级任务。



注意

为了获得最佳结果，请在生产环境中的不同服务器上安装 KIE 服务器和无头流程自动化管理器控制器。在开发环境中，您可以在同一个服务器上安装 KIE 服务器和无头进程 Automation Manager 控制器。

先决条件

- IBM WebSphere Application Server 实例已配置，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。
- 您有足够的用户权限来完成安装。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
2. 下载 **Red Hat Process Automation Manager 7.12.0 附加组件**。
3. 将下载的 **rhpm-7.12.0-add-ons.zip** 文件提取到临时目录中。
4. 将 **rhpm-7.12.0-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-7.12.0-add-ons/rhpm-7.12.0-controller-ee7.zip** 文件。
6. 重新打包 **controller.war** 目录：
 - a. 导航到 **TEMP_DIR/rhpm-7.12.0-add-ons/rhpm-7.12.0-controller-ee7/controller.war** 目录。
 - b. 选择 **TEMP_DIR/rhpm-7.12.0-add-ons/rhpm-7.12.0-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 Automation Manager 控制器的文件。
 - d. 可选：将新 **controller.war** 文件复制到从中部署更方便的位置。
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，再选择要上传的文件。
10. 选择 **Fast Path** 并点 **Next**。
此时会打开 **Install New Application** 向导。
11. 将 **Application Name** 更改为 **controller**，再单击 **Next**。
12. 根据您的具体要求，将无头进程 Automation Manager 控制器模块映射到服务器，然后点 **Next**。
13. 对于 **Message-Driven Beans** 的 **Bind Listeners**，为这两个 Bean 选择 **Activation Specification**，在 **Target Resource JNDI Name** 字段中输入 **jms/activation/KIE.SERVER.REQUEST**，再输入 **jms/cf/KIE.SERVER.REQUEST** JNDI 名称作为 **KIE.SERVER.REQUEST** 连接。
14. 在 **Map Virtual Hosts for Web Modules** 部分中，保留默认值并单击 **Next**。
15. 将上下文 **root** 设置为 **controller**。
16. 在 **Modules** 的 **Metadata** 中，保留默认值并单击 **Next**。
17. 点 **Finish** 安装无头进程 Automation Manager 控制器，然后点 **Save** 保存更改。

49.1. 为无头进程 AUTOMATION MANAGER 控制器配置类加载

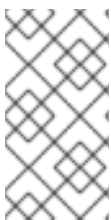
安装无头进程 Automation Manager 控制器后，您必须配置 KIE 服务器类加载，以便将父类设置为最后一个加载。

流程

1. 导航到 **Applications** → **Application Types** → **WebSphere Enterprise Applications**，再点 **kie-server**。
2. 在左侧的 **Details Properties** 标题下，单击 **Class Loading** 和 **Update Detection**。
3. 在属性中，将 **Class Loader Order** 更改为 **使用本地类加载器的类，首先（父）** 和 **WAR Class Loader Policy** 用于应用程序。
4. 保存对主配置的改变。

49.2. 为无头进程 AUTOMATION MANAGER 控制器设置系统属性

安装无头进程 Automation Manager 控制器后，在应用服务器或服务器上设置本节中列出的系统属性，以启用与无头进程 Automation Manager 控制器正确交互。



注意

为了获得最佳结果，请在生产环境中的不同服务器上安装 KIE 服务器和无头流程自动化管理器控制器。在开发环境中，您可以在同一个服务器上安装 KIE 服务器和无头进程 Automation Manager 控制器。在这两种情况下，请确保在安装了无头进程 Automation Manager 控制器的所有应用服务器上更改这些属性。

先决条件

- KIE 服务器和无头进程 Automation Manager 控制器安装在应用服务器实例上。

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上指定以下 JVM 属性值：

表 49.1. 无头进程 Automation Manager 控制器所需的属性

名称	要求
org.kie.server.user	具有 kie-server 角色的用户
org.kie.server.pwd	org.kie.server.user 属性中指定的用户密码

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 Automation Manager 控制器时，KIE Server 的必要属性

名称	要求
org.kie.server.controller.user	具有 kie-server 角色的用户
org.kie.server.controller.pwd	为 org.kie.server.controller.user 属性指定的用户的密码
org.kie.server.id	KIE 服务器安装的 ID 或名称，如 rhdm700-decision-server-1
org.kie.server.location	KIE 服务器的 URL、 http://<HOST>:<PORT>/kie-server/services/rest/server
org.kie.server.controller	无头进程 Automation Manager 控制器的 URL， http://<HOST>:<PORT>/controller/rest/controller

<HOST> 是 KIE Server 主机的 ID 或名称，如 **localhost** 或 **192.7.8.9**。

<PORT> 是 KIE Server 主机的端口号，如 **7001**。

49.3. 映射无头进程 AUTOMATION MANAGER 控制器组和角色

安装无头进程 Automation Manager 控制器后，在 WebSphere Integrated Solutions 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

- 无头进程 Automation Manager 控制器安装在 IBM WebSphere Application Server 实例中。

流程

1. 在 WebSphere Integrated Solutions 控制台中，创建 **kie-server** 组和该组的用户，如创建 [KIE Server 组和角色](#) 中所述。
2. 导航到 **Applications → Application Types → WebSphere Enterprise Applications**，然后选择新安装的无头进程 Automation Manager 控制器。
3. 在 **Details Properties** 下，点 **Security Role to User/Group Mapping**。
4. 选择 **kie-server** 角色，然后单击 **Map Groups** 以搜索 **kie-server** 组。
5. 将 **kie-server** 组从 **Available** 列表移到 **Selected** 列表中，再单击 **OK**。此映射可让之前创建的管理员用户访问进程 Automation Manager。
6. 点 **Save** 以完成映射。

49.4. 验证安装

安装无头进程 Automation Manager 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后，验证无头进程 Automation Manager 控制器是否正常工作。

先决条件

- KIE 服务器和无头进程 Automation Manager 控制器安装在应用服务器实例上。
- 您已为应用服务器上的无头进程 Automation Manager 控制器设置了所有必要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头进程 Automation Manager 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 Automation Manager 控制器主机的 ID 或名称，如 **localhost** 或 **192.7.8.9**。

<PORT > 是无头进程 Automation Manager 控制器主机的端口号，如 **7001**。

<CONTROLLER > 和 **<CONTROLLER_PWD >** 是您在本节中创建的用户凭证。

该命令应返回有关 KIE 服务器实例的信息。



注意

另外，您可以使用 KIE Server Java API 客户端来访问无头进程 Automation Manager 控制器。

如果无头进程 Automation Manager 控制器没有运行，请停止并重启应用程序服务器实例，并尝试访问无头进程 Automation Manager 控制器 URL 或 API。

第 50 章 在 IBM WEBSPHERE APPLICATION SERVER 中配置嵌入式流程引擎和决策引擎

嵌入式引擎是一个轻量级工作流和规则引擎，使您能够执行您的决策和业务流程。嵌入式引擎可以是 Red Hat Process Automation Manager 应用程序的一部分，或者通过 OpenShift、Kubernetes 和 Docker 部署为服务。您可以通过 API 或一组上下文和依赖项注入(CDI)服务将引擎嵌入到 Red Hat Process Automation Manager 应用程序中。

如果要将在嵌入式引擎与红帽流程自动化管理器应用程序一起使用，则必须通过将 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 Business 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>7.12.0.redhat-00008</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，定义了面向用户的产品依赖项的版本，因此您不需要指定这些 `<dependency>` 元素的 `<version>` 的子元素。但是，您必须使用 `<dependency>` 元素来声明要在项目中使用的依赖项。
 - 对于基本的 Red Hat Process Automation Manager 项目，请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，请声明以下依赖项：

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（如规则和进程定义）的 JAR 文件时，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 `<project>` 元素下的 **kjar** 打包类型。在以下示例中，**\${kie.version}** 是 Red Hat Process Automation Manager 和 Maven 库版本之间的映射中列出的 Maven 库版本？

```
<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>
```

- 如果您在项目中使用带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 文件复制 **version.org.hibernate-4ee7** 属性，在 **pom.xml** 文件的 **dependencyManagement** 部分中声明以下 hibernate 依赖项：

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。您应该加密控制器和 KIE 服务器密码。如果 Business Central 和 KIE 服务器部署到不同的应用服务器，则两个应用服务器将使用密钥存储。

为您的密钥存储使用 Java CryptographyExtension 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，KIE 服务器密码会以纯文本形式存储在系统属性中。

先决条件

- KIE 服务器安装在 IBM WebSphere 应用服务器中。
- 已创建具有 **kie-server** 角色的 KIE 服务器用户，如 [第 48.1 节 “创建 KIE Server 组和角色”](#) 所述。
- 安装了 Java 8 或更高版本。

流程

1. 创建 JCEKS 密钥存储。
2. 出现提示时，输入您创建的 KIE Server 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
kie.keystore.keyStoreURL	<KEYSTORE_URL>	要使用的 JCEKS 的 URL，例如 file:///home/kie/keystores/keystore.jceks
kie.keystore.keyStorePwd	<KEYSTORE_PWD>	JCEKS 的密码
kie.keystore.key.server.alias	<KEY_SERVER_ALIASES>	存储密码的 REST 服务的密钥别名
kie.keystore.key.server.pwd	<KEY_SERVER_PWD>	带有存储密码的 REST 服务别名的密码
kie.keystore.key.ctrl.alias	<KEY_CONTROL_ALIASES>	存储密码的默认 REST Process Automation Controller 的密钥别名
kie.keystore.key.ctrl.pwd	<KEY_CONTROL_PWD>	带有存储密码的默认 REST Process Automation Controller 的别名密码

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](#)
- [使用指导决策表设计决策服务](#)

部分 VI. 在 ORACLE WEBLOGIC 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 Oracle WebLogic 服务器，并在该 Oracle 服务器实例上安装 KIE 服务器。



注意

对 Oracle WebLogic 服务器上的红帽流程自动化管理器的支持现在处于维护阶段。红帽将继续在 Oracle WebLogic 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。
- 红帽只会发布具有严重影响和关键任务程序错误修复的合格安全补丁。

在未来，红帽可能会要求客户迁移到与红帽混合云策略兼容的新平台和产品组件。

先决条件

- 已安装 Oracle WebLogic Server 实例版本 12.2.1.3.0 或更高版本。有关完整的安装说明，请查看 [Oracle WebLogic Server 产品页面](#)。
- 您可以访问 Oracle WebLogic Server 管理控制台，通常在 **http://<HOST>:7001/console** 上。

第 53 章 KIE SERVER

KIE Server 是存储并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 REST、Java 消息服务(JMS)或 Java 客户端侧应用程序可用的接口来实例化和执行规则，以及通过解析程序管理流程、作业和红帽构建 OptaPlanner 功能。

创建为 Web 可部署的 WAR 文件，KIE 服务器可以部署到任何 Web 容器中。KIE 服务器的当前版本包括在 Red Hat Decision Manager 和 Red Hat Process Automation Manager 的默认扩展中。

KIE 服务器占用量较低，内存消耗较少，因此可以在云实例上轻松部署。此服务器的每个实例都可以打开和实例化多个容器，允许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 是一个 Java EE 应用服务器，它提供一组标准的 API 来创建分布式 Java 应用程序，可访问各种服务，如数据库、消息传递服务以及到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 安装并运行 ORACLE WEBLOGIC SERVER

必须安装并运行 Oracle WebLogic Server，才能应用容纳 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 域。

有关最新和详细安装说明，请查看 [Oracle WebLogic Server 产品页面](#)。



注意

如果您已经运行一个使用与您要启动的服务器使用的相同侦听器端口的 Oracle WebLogic 服务器实例，您必须在启动第二个服务器前停止第一个服务器。

流程

1. 从 Oracle WebLogic Server [Downloads](#) 页 下载 Oracle WebLogic Server 12.2.1.3.0 或更高版本。
2. 登录目标系统，并验证您的系统中是否已存在经过认证的 JDK。安装程序需要认证的 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“关于 Oracle Fusion 中间件安装的 JDK 要求”部分。
3. 进入您下载安装程序的目录。
4. 要启动安装程序，请从系统中的 JDK 目录中运行 **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 操作系统中输入以下命令：

```
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 位置替换为系统中的实际 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 **<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 操作系统中输入以下命令：

```
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, 替换以下位置拥有者 :

- 将 **<HOST >** 替换为主机服务器的系统名称或 IP 地址。
- 将 **& It;PORT >** 替换为主机服务器正在侦听请求的端口号 (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 Web 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** 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, 请输入您的管理凭据。

第 56 章 为 KIE 服务器配置 ORACLE WEBLOGIC 服务器

在使用 Oracle WebLogic 服务器部署 KIE 服务器之前，您必须在 Oracle WebLogic Server 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

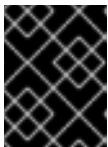
- Oracle WebLogic Server 已安装并运行。
- 您已登录到 WebLogic 管理控制台。

56.1. 配置 KIE SERVER 组和用户

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点 **New** 来创建新的安全域。
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已知标题没有冲突。例如，如果有一个名为 **kie-server** 的角色，则不要创建具有用户名 **kie-server** 的用户。

6. 单击新创建的用户，然后返回到 **组** 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 服务器中配置 JDBC 数据源

数据源是一种对象，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 服务器配置数据源，以确保服务器与指定数据库之间的相应数据交换。

通常，使用红帽流程自动化管理器的解决方案在单个事务中管理多个资源。JMS 用于异步作业、事件和计时器，例如：Red Hat Process Automation Manager 尽可能在数据源中有一个 XA 驱动程序，以确保数据原子性和结果一致。如果不同架构的事务代码存在于监听器内，或源自 jBPM 引擎提供的 hook，则还需要 XA 驱动程序。

除非您没有参与单个事务的多个资源，否则不要使用非 XA 数据源。

先决条件

- 您要用来创建数据库连接的 JDBC 驱动程序安装在您要在其上部署数据源的所有服务器上。某些 JDBC 驱动程序安装有 Oracle WebLogic Server，如用于 DB2、Informix、MS SQL Server 和

pkexec 的 WebLogic 品牌数据直接 JDBC 驱动程序。有关 JDBC 驱动程序的更多信息，请参阅在 Oracle 帮助中心 [中使用 JDBC 驱动程序和 WebLogic 服务器](#)。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - **产品**：流程自动化管理器
 - **Version**: 7.12
2. 下载 **Red Hat Process Automation Manager 7.12.0 附加组件**。
3. 完成以下步骤以准备数据库：
 - a. 将 **rhpm-7.12.0-add-ons.zip** 提取到临时目录中，如 **TEMP_DIR**。
 - b. 提取 **TEMP_DIR/rhpm-7.12.0-migration-tool.zip**。
 - c. 将您的当前目录更改为 **TEMP_DIR/rhpm-7.12.0-migration-tool/ddl-scripts** 目录。此目录包含多种数据库类型的 DDL 脚本。
 - d. 将数据库类型的 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

如果您将 PostgreSQL 或 Oracle 与 Spring Boot 结合使用，您必须导入相应的 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。在配置文件 (**config.xml**) 和管理控制台中每当引用此数据源时使用此名称。
 - **JNDI Name**：输入此 JDBC 数据源将被绑定的 JNDI 路径。应用在保留连接时通过此名称在 JNDI 树中查找数据源。
 - **Database Type**：选择您要连接到的数据库的 DBMS。如果没有列出您的 DBMS，请选择 **其他选项**。
8. 单击 **Next** 以继续。
9. 选择您要用于连接数据库的数据库驱动程序。列表包括所选 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** 页面上，保留选择 **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，以禁用此数据源中的全局事务（忽略）全局事务。在大多数情况下，应保留选择以优化数据效率的选项。
 - **2-Phase Commit**：选择这个选项来启用标准 XA 处理。这个选项只有在您选择 XA JDBC 驱动程序来创建数据库连接时才可用。
 - **logging Last Resource**：选择这个选项来启用非 XA JDBC 连接，以使用日志记录 Last Resource (LLR)事务优化来参与全局事务。建议将此选项代替 Emulate Two-Phase Commit。只有在您选择非 XA JDBC 驱动程序进行数据库连接时，此选项才可用。
 - **模拟两个Phase Commit**：选择这个选项来启用非 XA JDBC 连接，以使用 JTA 模拟参与分布式事务。只有在您的应用程序可以容忍 heuristic 条件时，才选择这个选项。只有在您选择非 XA JDBC 驱动程序进行数据库连接时，此选项才可用。
 - **One-Phase Commit**：选择这个选项来启用非 XA 连接，以参与作为唯一的事务参与者的全局事务。只有在您选择非 XA JDBC 驱动程序进行数据库连接时，此选项才可用。
11. 单击 **Next** 以继续。
12. 在 **Connection Properties** 页面中，输入以下属性的值：
 - **Service Name**：指定要连接的数据库的服务名称。如果提供了多个数据源，则每个数据源必须相同。只有在为 Oracle Real Application Clusters (RAC)选择了其中一个可用 service-instance 连接驱动程序时，此字段才可用。
 - **数据库名称**：输入您要连接到的数据库的名称。确切的数据库名称要求因 JDBC 驱动程序和 DBMS 而异。
 - **Host Name**：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，如果提供了多个数据源，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 Password**：输入数据库用户帐户的密码。
 - **Oracle.jdbc.DRCPConnectionClass**：可选，如果您的环境需要，请输入 Database Resident Connection Pooling (DCRP)连接类。
13. 单击 **Next** 以继续。
14. 在 **Test Database Connection** 页面上，检查连接参数，再单击 **Test Configuration**。Oracle WebLogic Server 尝试从管理服务器创建与数据库的连接。连接测试的结果会显示在页面的顶部。如果测试失败，请更正任何配置错误并重试测试。
15. 如果您在管理服务器上未安装 JDBC 驱动程序，请单击 **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. 返回到 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 帮助中心中的 [适用于 Oracle WebLogic 服务器的 JDBC 数据源](#)。

56.3. 配置 JAVA 消息服务(JMS)

Java 消息服务(JMS)是一种 Java API，KIE 服务器用于与其他应用服务器（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，以便与 KIE 服务器和 Oracle WebLogic Server 搭配使用。

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. 输入您的 JMS 服务器的名称，然后点 **Next**。
4. 选择为 KIE Server 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 **New** 以创建模块。
3. 输入模块名称并点 **Next**。
4. 选择为 KIE Server 部署选择的目标服务器，然后单击 **Finish**。
5. 单击新创建的模块名称，然后单击 **Subdeployments**。
6. 点 **New** 为您的模块创建子部署。
7. 为您的子部署指定一个名称，然后点 **Next**。
8. 选中复选框，以选择之前创建的 JMS 服务器。
9. 单击 **Finish** 以完成子部署配置。

56.3.3. 创建 JMS 连接工厂

要启用 KIE Server 的消息传递，您必须创建特定的 JMS 连接工厂来发送和接收消息。

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**，再单击 **Next**。
4. 对于下表中列出的每个必要连接工厂，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

名称	默认值	用于
KIE.SERVER.REQUEST	jms/cf/KIE.SERVER.REQUEST	将所有请求发送到 KIE 服务器
KIE.SERVER.RESPONSE	jms/cf/KIE.SERVER.RESPONSE	接收 KIE 服务器生成的所有响应
KIE.SERVER.EXECUTOR	jms/cf/KIE.SERVER.EXECUTOR	KIE Server executor 服务

5. 点 **Finish** 添加连接工厂，然后对每个所需的工厂重复此操作。

56.3.4. 创建 JMS 队列

JMS 队列是点对点消息传递的目的地端点。您必须创建特定的 JMS 队列，才能使用 KIE 服务器启用 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 **New** 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

名称	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 Server executor 服务

5. 选择与 JMS 服务器连接的 JMS 模块子部署。
6. 单击 **Finish** 以添加队列，然后对每个所需队列重复此操作。

56.4. 在 ORACLE WEBLOGIC SERVER 中设置系统属性

在部署 KIE 服务器前，设置本节中列出的系统属性。

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
```

如果您不增加 JVM 内存大小，Oracle WebLogic 服务器在部署 KIE 服务器时冻结或导致部署错误。

2. 在 Oracle WebLogic Server 实例中为 KIE 服务器指定以下系统属性：

表 56.3. KIE 服务器的系统属性

名称	值	描述
kie.server.jms.queues.response	jms/KIE.SERVER.RESPONSE	用于 KIE 服务器使用的响应的 JMS 队列的 JNDI 名称。
org.kie.server.domain	OracleDefaultLoginConfiguration	JAAS LoginContext 域，用于在使用 JMS 时对用户进行身份验证。
org.kie.server.persistence.ds	jdbc/jbpm	KIE 服务器的数据源 JNDI 名称。
org.kie.server.persistence.tm	org.hibernate.service.jta.platform.internal.WeblogicJtaPlatform	用于设置 Hibernate 属性的事务管理器平台。
org.kie.server.persistence.dialect	示例： org.hibernate.dialect.H2Dialect	指定要使用的 Hibernate dialect。根据数据源设置。

名称	值	描述
org.kie.executor.jms.queue	JMS/KIE.SERVER.EXECUTOR	KIE 服务器的作业执行程序 JMS 队列。
org.kie.executor.jms.cf	jms/cf/KIE.SERVER.EXECUTOR	KIE 服务器的作业执行者 JMS 连接工厂。
org.kie.server.router	示例： http://localhost:9000	(可选) 指定应用服务器是集群 KIE 服务器环境中的一个或多个 KIE 服务器路由器 (Smart routers) 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. 停止并重启 ORACLE WEBLOGIC SERVER

在 Oracle WebLogic Server 中配置了所有必要的系统属性后，停止并重启 Oracle 服务器以确保应用配置。

流程

1. 在 WebLogic 管理控制台中，导航到 **Change Center → Lock & Edit**。
2. 在 **Domain Structure** 下，点 **Environment → Servers → Control**。
3. 选择您要停止的服务器，然后单击 **Shutdown**。
4. 选择 **When Work Completes** 以正常关闭服务器，或者选择 **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** 完成关闭。
6. 关闭完成后，在命令行终端中进入域目录 **WLS_HOME/user_projects/<DOMAIN_NAME>**。例如：

```
WLS\user_projects\mydomain
```

7. 输入以下命令之一重启 Oracle WebLogic Server 以应用新配置：

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 服务器中配置了所有必要的系统属性后，您可以使用 Oracle WebLogic Server 安装 KIE Server，以简化红帽流程自动化管理器应用程序管理。

先决条件

- 如 [第 56 章 为 KIE 服务器配置 Oracle WebLogic 服务器](#) 所述配置 Oracle WebLogic 服务器实例。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
2. 下载 **适用于所有支持的 EE7 Containers 的 Red Hat Process Automation Manager 7.12.0 KIE Server**。
3. 将下载的 **rhpam-7.12.0-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 **rhpam-7.12.0-kie-server-ee7.zip** 文件的临时目录，然后前往 **rhpam-7.12.0-kie-server-ee7/kie-server.war**。
7. 选择 **kie-server.war** 文件，然后单击 **Next** 以继续。
8. 选择 **Install this deployment as a application as the targeting style** 并点 **Next**。
9. 将应用程序名称设置为 **kie-server**，并将安全模型设置为 **DD**。剩余的选项保留默认值，然后单击 **Next** 以继续。
10. 在 **Additional Configuration** 部分中，选择 **No**，**稍后查看配置** 并点 **Finish**。

57.1. 验证 ORACLE WEBLOGIC SERVER 上的 KIE 服务器安装

在 Oracle WebLogic Server 上安装 KIE Server 后，验证安装是否成功。

先决条件

- 如 [第 56 章 为 KIE 服务器配置 Oracle WebLogic 服务器](#) 所述配置 Oracle WebLogic 服务器实例。
- KIE Server 如 [第 57 章 使用 Oracle WebLogic Server 安装 KIE 服务器](#) 所述安装。

流程

1. 在网页浏览器中输入 KIE Server 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，请停止并重启 Oracle WebLogic 服务器实例，然后重新尝试访问 KIE 服务器 URL。

第 58 章 使用 ORACLE WEBLOGIC SERVER 安装并运行无头进程 AUTOMATION MANAGER 控制器

要使用 KIE Server REST API 或 Java 客户端 API 与 KIE 服务器交互，请使用 Oracle WebLogic Server 安装无头流程自动化管理器控制器。无头进程 Automation Manager 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 Automation Manager 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

- Oracle WebLogic Server 实例已配置，如 [第 56 章 为 KIE 服务器配置 Oracle WebLogic 服务器](#) 所述。
- KIE 服务器安装在 Oracle WebLogic Server 实例上。
- 您有足够的用户权限来完成安装。

流程

1. 导航到红帽客户门户网站中的 [Software Downloads](#) 页面（需要登录），然后从下拉菜单中选择产品和版本：
 - 产品：流程自动化管理器
 - Version: 7.12
2. 下载 [Red Hat Process Automation Manager 7.12.0 附加组件](#)。
3. 将下载的 `rhpmam-7.12.0-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 Automation Manager 控制器创建一个用户，如 **控制器**，以及这个新用户的密码，然后点 **OK**。有关创建组和用户的详情，请参考 [第 56.1 节 “配置 KIE Server 组和用户”](#)。
6. 前往 **Deployments** 以查看所有现有应用。
7. 点 **Install**。
8. 导航到您下载并提取 `rhpmam-7.12.0-add-ons.zip` 文件的临时目录，再前往 `rhpmam-7.12.0-add-ons/rhpmam-7.12.0-controller-ee7.zip/controller.war`。
9. 选择 **controller.war** 文件，再单击 **Next** 以继续。
10. 选择 **Install this deployment as a application as the targeting style** 并点 **Next**。
11. 将应用程序的名称保留为 **controller**，将安全模式设置为 **DD Only**。剩余的选项保留默认值，然后单击 **Next** 以继续。
12. 在 **Additional Configuration** 部分中，选择 **No**，稍后查看配置 并点 **Finish**。

58.1. 为无头进程 AUTOMATION MANAGER 控制器设置系统属性

安装无头进程 Automation Manager 控制器后，在应用服务器或服务器上设置本节中列出的系统属性，以启用与无头进程 Automation Manager 控制器正确交互。



注意

为了获得最佳结果，请在生产环境中的不同服务器上安装 KIE 服务器和无头流程自动化管理器控制器。在开发环境中，您可以在同一个服务器上安装 KIE 服务器和无头进程 Automation Manager 控制器。在这两种情况下，请确保在安装了无头进程 Automation Manager 控制器的所有应用服务器上更改这些属性。

先决条件

- KIE 服务器和无头进程 Automation Manager 控制器安装在应用服务器实例上。

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上指定以下 JVM 属性值：

表 58.1. 无头进程 Automation Manager 控制器所需的属性

名称	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 Automation Manager 控制器时，KIE Server 的必要属性

名称	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> 属性指定的用户的密码
<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 服务器的 URL、 <code>http://<HOST>:<PORT>/kie-server/services/rest/server</code>
<code>org.kie.server.controller</code>	无头进程 Automation Manager 控制器的 URL， <code>http://<HOST>:<PORT>/controller/rest/controller</code>

<HOST> 是 KIE Server 主机的 ID 或名称，如 `localhost` 或 `192.7.8.9`。

<PORT> 是 KIE Server 主机的端口号，如 `7001`。

58.2. 验证安装

安装无头进程 Automation Manager 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后，验证无头进程 Automation Manager 控制器是否正常工作。

先决条件

- KIE 服务器和无头进程 Automation Manager 控制器安装在应用服务器实例上。
- 您已为应用服务器上的无头进程 Automation Manager 控制器设置了所有必要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头进程 Automation Manager 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 Automation Manager 控制器主机的 ID 或名称，如 **localhost** 或 **192.7.8.9**。

<PORT> 是无头进程 Automation Manager 控制器主机的端口号，如 **7001**。

<CONTROLLER> 和 <CONTROLLER_PWD> 是您在本节中创建的用户凭证。

该命令应返回有关 KIE 服务器实例的信息。



注意

另外，您可以使用 KIE Server Java API 客户端来访问无头进程 Automation Manager 控制器。

如果无头进程 Automation Manager 控制器没有运行，请停止并重启应用程序服务器实例，并尝试访问无头进程 Automation Manager 控制器 URL 或 API。

第 59 章 在 ORACLE WEBLOGIC SERVER 中配置嵌入式流程引擎和决策引擎

嵌入式引擎是一个轻量级工作流和规则引擎，使您能够执行您的决策和业务流程。嵌入式引擎可以是 Red Hat Process Automation Manager 应用程序的一部分，或者通过 OpenShift、Kubernetes 和 Docker 部署为服务。您可以通过 API 或一组上下文和依赖项注入(CDI)服务将引擎嵌入到 Red Hat Process Automation Manager 应用程序中。

如果要将嵌入式引擎与红帽流程自动化管理器应用程序一起使用，则必须通过将 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 Business 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>7.12.0.redhat-00008</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，定义了面向用户的产品依赖项的版本，因此您不需要指定这些 `<dependency>` 元素的 `<version>` 的子元素。但是，您必须使用 `<dependency>` 元素来声明要在项目中使用的依赖项。
 - 对于基本的 Red Hat Process Automation Manager 项目，请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，请声明以下依赖项：

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（如规则和进程定义）的 JAR 文件时，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 `<project>` 元素下的 **kjar** 打包类型。在以下示例中，**`${kie.version}`** 是 Red Hat Process Automation Manager 和 Maven 库版本之间的映射中列出的 Maven 库版本？

```
<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>
```

- 如果您在项目中使用时带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 文件复制 **version.org.hibernate-4ee7** 属性，在 **pom.xml** 文件的 **dependencyManagement** 部分中声明以下 hibernate 依赖项：

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。您应该加密控制器和 KIE 服务器密码。如果 Business Central 和 KIE 服务器部署到不同的应用服务器，则两个应用服务器将使用密钥存储。

为您的密钥存储使用 Java CryptographyExtension KeyStore (JCEKS)，因为它支持对称密钥。使用 KeyTool (JDK 安装的一部分)来创建新的 JCEKS。



注意

如果没有使用 JCEKS 配置 KIE 服务器，KIE 服务器密码会以纯文本形式存储在系统属性中。

先决条件

- KIE 服务器安装在 Oracle WebLogic 服务器中。
- 已创建具有 **kie-server** 角色的 KIE 服务器用户，如 [第 56.1 节“配置 KIE Server 组 and 用户”](#) 所述。
- 安装了 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 >**：密钥存储将存储的路径
 - **<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 >**：进程条目的别名
2. 出现提示时，输入您创建的 KIE Server 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
kie.keystore.keyStoreURL	<KEYSTORE_URL>	要使用的 JCEKS 的 URL，例如 file:///home/kie/keystores/keystore.jceks
kie.keystore.keyStorePwd	<KEYSTORE_PWD>	JCEKS 的密码

系统属性	占位符	描述
kie.keystore.key.server.alias	<KEY_SERVER_ALI AS>	存储密码的 REST 服务的密钥别名
kie.keystore.key.server.pwd	<KEY_SERVER_PWD >	带有存储密码的 REST 服务别名的密码
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>	带有存储密码的默认 REST Process Automation Controller 的别名密码

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](#)
- [使用指导决策表设计决策服务](#)

部分 VII. 修补并升级 RED HAT PROCESS AUTOMATION MANAGER 7.12

作为 Red Hat Process Automation Manager 用户，您可以在红帽客户门户网站中提供时对 Red Hat Process Automation Manager 应用更新，以便使 Red Hat Process Automation Manager 发行版本保持最新功能增强和修复。红帽为新产品版本提供了更新工具和产品通知，以便您能够更好地将有用的更新应用到红帽流程自动化管理器和 KIE 服务器。

先决条件

- 您有红帽客户门户网站帐户。
- 安装了 Red Hat Process Automation Manager 7.12。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，并且希望升级到 Red Hat Process Automation Manager 7.12，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 版本指定为数字 **Major.Minor.Patch** 格式，如 **7.11.1**。在本例中，主发行版本是 **7.x.x**，次发行版本为 **7.11.x**，补丁版本为 **7.11.1**。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发行版本移到另一个主发行版本（如从 Red Hat JBoss BPM Suite 6.4.x 到 Red Hat Process Automation Manager 7.0）时，通常会需要数据迁移。自动化迁移工具通常随 Red Hat Process Automation Manager 的新主要版本提供，以协助迁移，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。

有关迁移说明示例，[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 的次发行版本包括功能增强、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 7.10.x 移到 7.11。自动更新工具通常由 Red Hat Process Automation Manager 的补丁更新和新的次版本提供，以便更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE Server 和无头流程 Automation Manager 控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）作为每个次发行版本的新工件发布，您必须重新安装它们以应用更新。

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

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该产品的所有以前发布的补丁更新，因此您不需要逐步应用每个补丁更新以应用最新的更新。例如，您可以将 Red Hat Process Automation Manager 7.10.0 或 7.10.1 更新至 Red Hat Process Automation Manager 7.10.2。但是，为了获得最佳红帽流程自动化管理器性能，请在产品更新可用时应用它们。有时，红帽可能会在现有产品普通更新周期外发布非计划补丁更新。它们可能包括 Red Hat 全球支持服务(GSS)提供的安全性或其他更新来修复特定问题，且可能不是累积更新。

自动更新工具通常由 Red Hat Process Automation Manager 的补丁更新和新的次版本提供，以便更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE Server 和无头流程 Automation Manager 控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）作为每个次发行版本的新工件发布，您必须重新安装它们以应用更新。

为确保在发行版本之间实现最佳过渡，并保持红帽流程自动化管理器最新发布，在红帽客户门户网站中提供新产品版本及 Red Hat Process Automation Manager 的更新。考虑在红帽客户门户网站中启用产品通知。

有关启用产品通知的详情，请参考 [第 63 章 在红帽客户门户网站中启用 Red Hat Process Automation Manager 通知](#)。

第 63 章 在红帽客户门户网站中启用 RED HAT PROCESS AUTOMATION MANAGER 通知

您可以在红帽客户门户网站中启用通知来接收 Red Hat Process Automation Manager 更新和公告。这些通知会通知您更新或添加与 Red Hat Process Automation Manager 相关的文档、产品版本和补丁更新。启用通知后，您可以更多地将产品更新应用到 Red Hat Process Automation Manager，因为红帽客户门户网站中的可用，以使您的 Red Hat Process Automation Manager 发行版本保持最新功能增强和修复。

先决条件

- 您有一个红帽客户门户网站帐户并已登录。

流程

1. 在红帽客户门户网站窗口右上角，点击您的配置集名称并点 **Notifications**。
2. 选择 **通知** 选项卡，然后单击 **Manage Notifications**。
3. 在 **Follow** 旁边，从下拉菜单中选择 **Products**，然后从出现的下一个下拉菜单中选择 **Red Hat Process Automation Manager**。
4. 单击 **Save Notification** 以完成。

您可以像相同的方式为任何其他产品添加通知。

第 64 章 将补丁更新和次版本的升级应用到 RED HAT PROCESS AUTOMATION MANAGER

自动更新工具通常由 Red Hat Process Automation Manager 的补丁更新和新的次版本提供，以便更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE Server 和无头流程 Automation Manager 控制器。其他 Red Hat Process Automation Manager 工件（如决策引擎和独立 Business Central）作为每个次发行版本的新工件发布，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版本升级到 Red Hat Process Automation Manager 7.12。Red Hat Process Automation Manager 的补丁更新，如从版本 7.12 更新至 7.12.1 的更新，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 7.10.x 升级到 7.11，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 的更新只包括在 Red Hat Process Automation Manager 更新工具中。必须使用红帽 JBoss EAP 补丁分发来应用红帽 JBoss EAP 的更新。有关 Red Hat JBoss EAP 补丁的更多信息，请参阅 [Red Hat JBoss EAP 补丁和升级指南](#)。

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 Server 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），然后从下拉菜单选择产品和版本。
如果您要升级到 Red Hat Process Automation Manager 的新次版本，如从版本 7.10.x 升级到 7.11，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 **rhcam-\$VERSION-update.zip** 文件提取到一个临时目录中。
此更新工具会自动更新 Red Hat Process Automation Manager 的某些组件，如 Business Central、KIE Server 和无头流程 Automation Manager 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 发行版本相关的任何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更新的文件。
当在 **blacklist.txt** 文件中列出文件时，更新脚本不会将文件替换为新版本，而是保留该文件，在同一位置会将新版本添加到 **.new** 后缀。如果您阻止不再分发的文件，则更新工具会创建一个带有 **.removed** 后缀的空标志文件。然后，您可以选择保留、合并或删除这些新文件。

blacklist.txt 文件中要排除的示例文件：

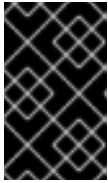
```
WEB-INF/web.xml // Custom file
styles/base.css // Obsolete custom file kept for record
```

更新后被阻止的文件内容：

```
$ ls WEB-INF
web.xml web.xml.new
```

```
$ ls styles
base.css base.css.removed
```

- 在命令终端中，导航到您提取 **rhpam-\$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，请确保您的红帽流程自动化管理器和 KIE 服务器实例没有运行。在运行 Red Hat Process Automation Manager 或 KIE Server 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 部分是相关发布目录的路径，**\$DISTRO_TYPE** 部分是您更新的分发类型。

Red Hat Process Automation Manager 更新工具支持以下发布类型：

- **RHPAM-business-central-eap7-deployable** : Updates Business Central (**business-central.war**)
- **RHPAM-kie-server-ee8** : 更新 KIE 服务器(**kie-server.war**)
- **RHPAM-kie-server-jws** : 在红帽 JBoss Web 服务器上更新 KIE 服务器(**kie-server.war**)
- **RHPAM-controller-ee7** : 更新无头进程 Automation Manager 控制器(**controller.war**)
- **RHPAM-controller-jws** : 更新 Red Hat JBoss Web Server (**controller.war**)上的无头进程 Automation Manager 控制器。

在 Red Hat JBoss EAP 上为完整的红帽流程自动化管理器分发的 Business Central 和 KIE Server 更新示例：

```
$ ./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
```

如果使用，对无头进程的 Automation Manager 控制器更新示例：

```
$ ./apply-updates.sh ~EAP_HOME/standalone/deployments/controller.war rhpam-
controller-ee7
```

更新脚本会在提取的 **rhpm-\$VERSION-update** 文件夹中创建一个 **备份** 文件夹，其中包含指定的发布副本，然后继续进行更新。

5. 更新工具完成后，在下载更新工具时返回到红帽客户门户网站的 **Software Downloads** 页面，并安装与 Red Hat Process Automation Manager 发行版本相关的任何其他更新或新的发行工件。对于 Red Hat Process Automation Manager 分发中已存在的文件，如适用于决策引擎或其他附加组件的 **.jar** 文件，请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 **Red Hat Process Automation Manager 7.12.0 Maven Repository** 工件 (**rhpm-7.12.0-maven-repository.zip**)，如在 **air-gap** 环境中下载 **Red Hat Process Automation Manager 7.12.x Maven Repository**，并将下载的 **rhpm-7.12.x-maven-repository.zip** 文件提取到现有的 **~/maven-repository** 目录中。
Maven 存储库更新示例：

```
$ unzip -o rhpm-7.12.x-maven-repository.zip 'rhba-7.12.1.GA-maven-repository/maven-repository/*' -d /tmp/rhbaMavenRepoUpdate
```

```
$ mv /tmp/rhbaMavenRepoUpdate/rhba-7.12.0.GA-maven-repository/maven-repository/$REPO_PATH/
```



注意

您可以在完成更新后删除 **/tmp/rhbaMavenRepoUpdate** 文件夹。

7. 可选：如果要将在 Red Hat Process Automation Manager 从使用基于属性的用户存储改为基于文件的用户存储，请完成以下步骤：
 - a. 导航到 **\$JBOSS_HOME** 目录，再运行以下命令之一：
 - 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=rhpm-$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=rhpm-$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 **rhpm-\$VERSION-update.zip** 文件的目录，再运行以下命令之一来应用 **kie-fs-realm** 补丁：

- 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. 完成应用所有相关更新后，启动 Red Hat Process Automation Manager 和 KIE Server 并登录到 Business Central。
9. 验证 Business Central 中是否存在和准确，在 Business Central 窗口右上角点击您的资料名称，然后单击 **About** 以验证更新的产品版本号。
如果您遇到错误或发现 Business Central 中的任何缺失的数据，您可以在 **rhpm-\$VERSION-update** 文件夹的备份文件夹中恢复内容，以恢复更新工具更改。您还可以在红帽客户门户网站中从之前版本的 Red Hat Process Automation Manager 中重新安装相关的发行工件。恢复之前的发行版本后，您可以尝试运行更新。

第 65 章 其他资源

- [规划 Red Hat Process Automation Manager 安装](#)
- [修补并升级 Red Hat Process Automation Manager 7.10](#)
- [Red Hat Process Automation Manager 7.12 发行注记](#)

附录 C. 版本控制信息

文档最新更新于 2023 年 2 月 1 日（周三）。

附录 D. 联系信息

Red Hat Process Automation Manager 文档团队 : brms-docs@redhat.com