Red Hat Single Sign-On 7.6

升级指南

适用于 Red Hat Single Sign-On 7.6

Last Updated: 2024-06-05

Red Hat Single Sign-On 7.6 升级指南

适用于 Red Hat Single Sign-On 7.6

法律通告

摘要

本书介绍了从以前的 Red Hat Single Sign-On 7.6 版本升级应用的指南。

目录

使开源包含更多 3

第 1 章 升级红帽单点登录 4
1.1. 关于升级 4
1.2. 将 KEYCLOAK 迁移到 RH-SSO 4

第 2 章 特定于发行的更改 6
2.1. RH SSO 7.6 6
2.2. RH SSO 7.5 12
2.3. RH SSO 7.4 17
2.4. RH-SSO 7.3 21
2.5. RH-SSO 7.2 25
2.6. RH-SSO 7.1 27

第 3 章 升级 RED HAT SINGLE SIGN-ON 服务器 29
3.1. 执行次要升级 29
3.2. 执行微小升级 47

第 4 章 升级 RED HAT SINGLE SIGN-ON 适配器 56
4.1. 与旧的适配器兼容 56
4.2. 升级 EAP 适配器 56
4.3. 升级 JAVASCRIPT 适配器 57
4.4. 升级 NODE.JS 适配器 57

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅我们的首席技术官 Chris Wright 提供的消息。

第 1 章 升级红帽单点登录

Red Hat Single Sign-On(RH-SSO)7.6 基于 Keycloak 项目，提供 Web 应用程序的安全性，具体根据 SAML 2.0、OpenID Connect 和 OAuth 2.0 等常用标准提供 Web 单点登录功能。Red Hat Single Sign-On Server 可以充当基于 SAML 或 OpenID Connect 的用户身份供应商，使用基于标准的令牌将企业用户目录或第三方 SSO 供应商与第三方 SSO 供应商进行关联。

RH-SSO 提供两种工作模式：独立服务器或受管域。单机服务器工作模式代表运行 RH-SSO 作为单一服务器实例。受管域工作模式允许从单一控制点管理多个 RH-SSO 实例。升级过程会根据实施的操作模式的不同而有所不同。如果适用，提供了每个模式的具体说明。

本指南的目的是记录成功从 Red Hat Single Sign-On 7.x 升级到 Red Hat Single Sign-On 7.6 所需的步骤。

1.1. 关于升级

根据您的 RH-SSO 版本，您选择三种升级类型之一。但是，如果您从 Keycloak 开始，请选择 此流程。

1.1.1. 主要升级

当 RH-SSO 从一个主要版本升级到另一个主版本（例如，从 Red Hat Single Sign-On 7.2 升级到 Red Hat Single Sign-On 8.0 时，需要进行一个主要升级或迁移。在可能需要重新编写应用程序或服务器扩展部分的主要版本之间可能会有 API 更改。

1.1.2. 小更新

Red Hat Single Sign-On 定期提供点版本，这些版本是包括错误修复、安全修复和新功能的次要更新。如果您计划从一个红帽单点登录版本升级到另一个红帽单点登录（例如，从 Red Hat Single Sign-On 7.3 升级到 Red Hat Single Sign-On 7.6），只要没有使用私有、不支持或技术预览 API，则应用程序或自定义服务器扩展不需要代码更改。

1.1.3. 微更新

Red Hat Single Sign-On 7.6 还定期提供包含程序错误和安全修复的微版本。微版本根据最后一个数字递增次要版本，例如从 7.6.0 增加到 7.6.1。这些版本不需要迁移，且不会影响服务器配置文件。ZIP 安装的补丁管理系统可以回滚补丁和服务器配置。

微版本仅包含更改的工件。例如，如果 Red Hat Single Sign-On 7.6.1 包含对服务器和 JavaScript 适配器的更改，则仅发布服务器和 JavaScript 适配器，需要更新。

1.2. 将 KEYCLOAK 迁移到 RH-SSO

您可以从 community 项目将受支持的红帽产品迁移到 Red Hat Single Sign-On。

先决条件

- 要在升级前了解新功能，请查看 更改。
- 验证您已将正确的 Keycloak 版本安装为起点。要迁移到 Red Hat Single Sign-On 7.6，首先安装 Keycloak 18.0.0。

流程

1. 执行 次版本升级步骤。虽然这个步骤有标签为 **Minor Upgrade**，但在这个迁移中会使用相同的步骤。

2. 执行 适配器升级步骤。

第 2 章 特定于发行的更改

在升级前仔细检查这些更改。

2.1. RH SSO 7.6

从 Red Hat Single Sign-On 7.5 到 Red Hat Single Sign-On 7.6 有以下更改。

2.1.1. 步骤验证

分步身份验证是一项新功能。此功能提供了 **cr** 客户端范围，它包含应当在令牌中添加 **cr** 声明的协议映射程序。现在，这个声明会被自动添加，而是使用这个客户端范围和协议映射程序。

客户端范围被添加为域"默认"客户端范围，因此将添加到所有新创建的客户端。出于性能考虑，在迁移期间，客户端范围不会自动添加到所有现有客户端。迁移后，客户端默认不会 具有cr 声明。考虑以下可能的操作：

- **如果您没有计划使用步骤身份验证功能，但您依赖令牌中的 acr 声明，您可以禁用 step_up_authentication 功能，如 服务器安装和配置指南 中所述。在进行正常身份验证时，声明将加上值 1，在 SSO 身份验证时将加上值 0。**

- **通过 admin REST API 或 admin 控制台手动向客户端 添加 关键客户端范围。这在您想要使用步骤验证时才需要。如果您在域中有大量客户端，并希望为它们使用 cr 声明， 您可以针对您的 DB 触发一些类似于您的 DB 的 SQL。但是，如果已经启动 Red Hat Single Sign-On，请记住清除缓存或重启服务器：**

```
insert into CLIENT_SCOPE_CLIENT (CLIENT_ID, SCOPE_ID, DEFAULT_SCOPE) select
CLIENT.ID as CLIENT_ID, CLIENT_SCOPE.ID as SCOPE_ID, true as DEFAULT_SCOPE
from CLIENT_SCOPE, CLIENT where CLIENT_SCOPE.REALM_ID='test' and
CLIENT_SCOPE.NAME='acr' and CLIENT.REALM_ID='test' and CLIENT.PROTOCOL='openid-
connect';
```

2.1.2. OpenID Connect Logout

以前版本的 Red Hat Single Sign-On 支持自动注销用户，并通过打开注销端点 URL（如 http(s)://example-host/auth/realms/my-realm-name/protocol/openid-connect/logout?redirect_uri=encodedRedirectUri ）来重定向到应用程序。虽然这种实施非常容易使用，但可能会对性能和安全性造成负面影响。根据 OpenID Connect RP-Initiated Logout 规格，新版本可以更好地支持注销。参数 redirect_uri 不再被支持；此外，新版本中，用户需要确认注销。当包含参数 post_logout_redirect_uri 和用于登录的 ID Token 时，可以省略确认并自动重定向到应用程序。

现有部署会受到以下方法影响：

- 如果您的应用程序直接使用到使用 redirect_uri 参数进行注销端点的链接，您可能需要更改如上所述。请考虑可以完全删除 redirect_uri 参数，或使用 id_token_hint 和 post_logout_redirect_uri 参数替换。

- 如果您使用 java 适配器，且应用程序会调用 httpServletRequest.logout（） 进行注销，则您不受影响，因为这个调用使用了注销端点的 backchannel 变体，且没有更改。

- 如果您使用最新的 javascript 适配器，您也不受影响。但是，如果您的应用程序使用 JavaScript 适配器的旧版本，您会受到影响，因为此适配器使用了已弃用的 redirect_uri 参数的注销端点变体。在这种情况下，您可能需要升级到 JavaScript 适配器的最新版本。

- 对于 Node.js 适配器，与 JavaScript 适配器相同。建议您更新到最新版本，因为较老的适配器版本使用已弃用的 redirect_uri 参数。使用最新的 Node.js 适配器时，只要您根据 /logout URL 使用注销 URL 或 Node.js 适配器示例所述，您不会受到这个安全漏洞的影响。但是，当应用程序直接使用 method keycloak.logoutUrl 时，您可以考虑将 idTokenHint 添加为此方法的第二个参数。在这个版本中，在这个版本中添加了 idTokenHint 作为第二个参数。idTokenHint 需要是登录期间获得的有效 ID 令牌。添加 idTokenHint 是可选的，但如果省略它，则您需要确认注销屏幕，如前面所述。另外，在注销后，它们不会重定向到应用程序。

有一个后向兼容选项，它允许您的应用程序仍然使用 redirect_uri 参数的旧格式。

您可以通过在 standalone-*.xml 文件中包括以下配置来启用这个参数

```
<spi name="login-protocol">
  <provider name="openid-connect" enabled="true">
    <properties>
      <property name="legacy-logout-redirect-uri" value="true"/>
    </properties>
  </provider>
</spi>
```

使用这个配置，您仍然可以将格式与 redirect_uri 参数搭配使用。请注意，如果省略 id_token_hint，则需要确认屏幕。

警告

以后的一些版本中将删除后向兼容开关。建议您尽可能立即更新您的客户端，而不是依赖这个交换机。

2.1.3. 删除 upload-scripts 功能

以前版本的 Red Hat Single Sign-On 支持通过管理控制台和 REST API 等管理界面来管理 JavaScript 代码。从此版本开始，您现在应该将脚本部署到服务器，以配置以下供应商：

- OpenID Connect Scriptmapper
- 脚本身份验证器（身份验证执行）
- JavaScript 策略

有关如何将脚本部署到服务器的详细信息，请参阅文档。请注意，要使用脚本，您仍需要启用 脚本 技术预览功能。

```
./standalone.sh -Dkeycloak.profile=preview
```

在部署脚本时，服务器将自动创建对应的供应商，以便您可以配置身份验证流程、映射程序和授权策略时选择它们。

通常，更新您的域的步骤如下：

- 在升级前，删除您要使用的任何脚本供应商。
- 升级后，请按照 文档 中的说明部署脚本。
-

更新身份验证流程、映射程序和客户端授权设置，以使用从部署到服务器的脚本创建的提供程序。

2.1.4. 帐户控制台模式升级

Patternfly(PF)React 库已被更新，@patternfly/react-core 从 v3.153.3 升级到 v4.147.0, @patternfly/react-icons from v3.15.16，将 @patternfly/react-styles 从 v3.7.14 变为 v4.11.8。进行了几个 UI 更新，使帐户控制台与 PF 设计标准一致。

自定义开发的帐户 UI 可能会因为 PF 中的破坏更改而与这些更新不兼容。大多数有问题的更改应该通过更新 PF 组件来改变。

resources:

- [Patternfly docs](https://www.patternfly.org)

已知有破坏更改的组件：

- 警报
- action prop 改为 actionClose
- 可扩展
- 重命名为 ExpandableSection
- 标题
- 大小 attr 现在使用 TitleSizes
- DataListContent

- noPadding 更改为 具有NoPadding
- Grid, Stack, Level, Gallery
- gutter attr 更改为 hasGutter
- 模态
- 从中更改大小控制（例如，isLarge ）以使用 ModalVariant，例如 ModalVariant = {ModalVariant.large}
- 选择
- ariaLabelTypeAhead to typeAheadAriaLabel
- 正在展开为 "展开 "
- ariaLabelledBy to aria-labelledby
- DataListContent
- NoPadding to hasNoPadding

2.1.5. 客户端策略迁移：客户端范围

如果您使用了策略，包括客户端范围条件并直接编辑的 JSON 文档，则需要在 JSON 文档中将 "scope" 字段名称更改为"scopes"。

2.1.6. Liquibase 升级到 4.6.2 版本

Liquibase 从 3.5.5 版本更新至 4.6.2，其中包括一些程序错误、一些程序错误修复以及使用

ServiceLoader 注册自定义扩展的新方法。

遵循 升级指南，特别是在升级前备份现有数据库。虽然我们最好测试 Liquibase 升级的结果，但有些安装可能会对我们使用特定的设置未知。

2.1.7. Red Hat Single Sign-On Operator 中的已弃用的功能

在这个版本中，我们已在 Red Hat Single Sign-On Operator 的 Keycloak CR 中弃用了 podDisruptionBudget 字段。当 Operator 部署到 OCP 4.12 及更高版本中时，此可选字段将被忽略。

作为临时解决方案，您可以在集群中创建 Pod Disruption Budget，例如：

```
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  labels:
    app: keycloak
  name: keycloak
spec:
  maxUnavailable: 1
  selector:
    matchLabels:
      component: keycloak
```

另请参阅 Kubernetes 文档。

2.1.8. Red Hat Single Sign-On Operator 中的关键程序错误修复

由于之前版本的 Operator 中存在一个关键错误，RH-SSO StatefulSet 上的 Selector 字段配置错误。错误配置可能会破坏升级过程从 7.5 到 7.6，从而会阻止成功进行 RH-SSO 部署。

随着 Operator 补丁发行版本，我们引入了一个修复程序。请注意，作为修复 Operator 的一部分，在从 7.5 升级到 7.6 的过程中可能会 删除并重新创建 RH-SSO StatefulSet。要使修复正常工作，请确定您使用 重新创建 的升级策略。请参阅 服务器安装和配置指南中的相关章节。

2.1.9. 使用 Red Hat Single Sign-On Operator 7.6.2 时探测的更改

为了与 7.6.2 中引入的 Red Hat Single Sign-On OpenShift 镜像中的更改一致，Operator 现在利用镜像中默认的存活度和就绪度探测，而不是使用自定义探测。对于现有的 Red Hat Single Sign-On 部署，Operator 将在升级过程中尝试自动更新探测。但是，如果您通过手动更改 keycloak-probes

ConfigMap 来自定义探测，Operator 不会更新探测以避免覆盖用户修改。在这种情况下，您必须手动更新探测（或删除 ConfigMap 以便 Operator 重新创建它）；否则升级的 Red Hat Single Sign-On 7.6.2 部署将显示为未就绪。

2.1.10. 在使用 Red Hat Single Sign-On Operator 7.6.5 时，会更改探测

为了允许在启用了 FIPS 的环境中运行，探测验证哈希算法已改变。对于探测的默认超时为 1 秒的基于模板的安装，并且指定了小于 1 的 CPU 限值，这可能会导致持续探测失败。如果这些失败会导致重启，您应该通过更改 DeploymentConfig 或实际到新发布的模板来增加探测超时，这与 Operator 中使用的内容有更大的超时值。

2.1.11. 不建议在帐户控制台版本 1 中使用步骤验证

帐户控制台 V1 与步骤身份验证相关的限制。问题是，用户可以使用密码对帐户控制台版本 1 进行身份验证，然后将 TOTP 凭证添加到用户或删除现有的 TOTP 凭证。这种情况意味着，管理给用户密码的任何人都可以通过添加另一个 TOTP 来绕过用户的第二因素身份验证。帐户控制台版本 2 不会出现此问题（如 Red Hat Single Sign-On 7.6.8 支持）。该版本始终强制用户与要添加或删除相应级别的凭证进行身份验证。

最好在使用步骤验证时避免使用帐户控制台版本 1。如果帐户主题明确切换到 keycloak 主题，则帐户控制台版本 1 可以正常工作。

2.2. RH SSO 7.5

以下已从 Red Hat Single Sign-On 7.4 到 Red Hat Single Sign-On 7.5 造成的。

2.2.1. 升级到 EAP 7.4

Red Hat Single Sign-On 服务器已升级，以使用 EAP 7.4 作为底层容器。这个更改不会直接涉及任何特定的红帽单点登录服务器功能，但与迁移相关的一些变化。

2.2.1.1. 依赖项更新

依赖项已更新至 EAP 7.4 服务器使用的版本。例如，Infinispan 组件版本现在是 11.0。

2.2.1.2. 配置更改

standalone（-ha.xml 和 domain.xml 文件中有一些配置更改。您应该遵循 第 3.1.2 节 “升级 Red Hat Single Sign-On 服务器” 部分自动处理配置文件迁移。

2.2.1.3. SmallRye manual change

当 standalone.xml 包含对 SmallRye 模块的引用时，需要手动更改。这些模块已从底层 JBoss EAP 发行版本中删除，如果配置引用它们，则服务器不会启动。在对配置进行任何更改之前，通过 migrate-standalone.cli 进行服务器配置迁移会失败。

要解决这个问题，请删除引用 SmallRye 模块的所有行。在默认配置中，您需要专门删除以下行：

```
<extension module="org.wildfly.extension.microprofile.config-smallrye"/>
<extension module="org.wildfly.extension.microprofile.health-smallrye"/>
<extension module="org.wildfly.extension.microprofile.metrics-smallrye"/>
```

```
<subsystem xmlns="urn:wildfly:microprofile-config-smallrye:1.0"/>
<subsystem xmlns="urn:wildfly:microprofile-health-smallrye:2.0" security-enabled="false" empty-liveness-checks-status="${env.MP_HEALTH_EMPTY_LIVENESS_CHECKS_STATUS:UP}" empty-readiness-checks-status="${env.MP_HEALTH_EMPTY_READINESS_CHECKS_STATUS:UP}"/>
<subsystem xmlns="urn:wildfly:microprofile-metrics-smallrye:2.0" security-enabled="false" exposed-subsystems="*" prefix="${wildfly.metrics.prefix:wildfly}"/>
```

2.2.1.4. 跨数据中心复制更改

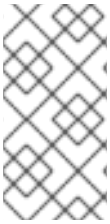
- 您需要将 RHDG 服务器升级到 8.x 版本。旧版本可能仍然可以正常工作，但无法保证，因为它已不再被测试。

- 我们建议您使用在配置 Infinispan 缓存时添加到 remote-store 元素中的 protocolVersion 属性。连接到 RHDG 服务器 8.x 时，推荐的热线协议版本是 2.9。Infinispan 库版本与红帽单点登录服务器和 RHDG 服务器之间的不同。如需了解更多详细信息，请参阅 Cross-Datacenter 文档。

- 建议您在 connection infinispan 子系统下使用 remoteStoreSecurityEnabled 属性。如需了解更多详细信息，请参阅 Cross-Datacenter 文档。

2.2.2. UserModel Migration

UserModel 包括某些字段、用户名、电子邮件、firstName 和 lastName，这些字段现在转换为自定义属性。进行了此更改，准备将更复杂的用户配置文件添加到 Red Hat Single Sign-On。



注意

如果数据库包含具有该名称的自定义属性的用户，这些属性将不再从数据库读取，并可删除。因此，在升级到 RH SSO 7.5 之前，请重命名与其中一个名称匹配的自定义属性。

这种情况意味着，用户名 现在可以被 UserModel.getFirstAttribute(UserModel.USERNAME) 访问和设置。其他字段存在类似的影响。SPIs 子类直接或间接处理 UserModel 的实施器应确保 setUsername 和 setSingleAttribute(UserModel.USERNAME, ...) 之间的行为一致。

策略评估功能的用户如果他们在评估中使用属性数量，则应该调整其策略。每个用户现在默认具有四个新属性。

UserModel 的公共 API 没有改变。不需要更改 frontend 资源或 SPI 访问用户数据。另外，数据库还没有改变。

2.2.3. 升级到 PatternFly 4

Red Hat Single Sign-On 登录主题组件已升级至 PatternFly 4。旧 PatternFly 3 可同时使用新的同时运行，因此可以保持 PF3 组件。

但是，对登录主题的设计进行了一些更改。请将您的自定义登录名升级到新版本。在 example /themes/theme/sunrise 目录中可以找到包含必要更改的示例。不需要额外的设置。

2.2.4. 用于 Instagram IdP 的新 API

Instagram IdP 现在使用一个新的 API。旧的旧的 API 已被弃用。此更改需要新的 API 凭证。有关详细信息，请参阅《服务器管理指南》。

对于使用 Instagram IdP 登录 Instagram 的用户，这些用户需要不同的身份验证方法，如密码。他们可以登录以手动更新其 Instagram 社交链接或在 Red Hat Single Sign-On 中创建一个新帐户。这个限制存在，因为上一个 API 中的 Instagram 用户 ID 与新的 API 不兼容。但是，新 API 会临时返回新的用户和旧用户 ID，以允许迁移。当用户登录时，Red Hat Single Sign-On 会自动迁移 ID。

2.2.5. SSRF 保护的有效请求 URI

如果使用 OpenID Connect 参数 request_uri，您的客户端需要配置 Valid Request URI，以防止 SSRF 攻击

您可以通过客户端详情页面上的 Admin Console 或管理员 REST API 或客户端注册 API 配置此功能。有效的 Request URIs 需要包含为特定客户端允许的 Request URI 值列表。

您可以使用通配符或相对路径，如 Valid Redirect URIs 选项。但是，我们建议在安全中使用 作为具体值。

2.2.6. 只读用户属性

现在，只读用户属性可用。在使用 REST API 更新用户或红帽单点登录用户界面时，用户或管理员不会编辑其中的一些用户属性。特别是，在使用以下任一时，这个更改很重要：

- 自定义用户存储供应商
- 自定义验证器
- 基于用户属性建立授权的自定义 JavaScript 授权策略
- X.509 身份验证器，包含自定义属性，用于将 X.509 证书映射到用户身份
- 任何其它自定义功能，其中部分用户属性都用作存储身份验证/授权/身份上下文的元数据，而不是简单的用户配置集信息。

详情请查看 Threat model 缓解方案。

2.2.7. Docker 身份验证后不需要用户会话

在使用 Docker 协议成功进行身份验证后，不会创建用户会话。有关详细信息，请参阅《服务器管理指南》。

2.2.8. 客户端凭证授予没有默认刷新令牌

对于此 Red Hat Single Sign-On 版本，OAuth2 客户端凭证Grant 端点默认不会发布刷新令牌。此行为与 OAuth2 规范一致。

因此，在成功进行客户端凭证身份验证后，Red Hat Single Sign-On 服务器端不会创建用户会话。结果提高了性能和内存消耗。我们鼓励使用 Client Credentials Grant 的客户端停止使用刷新令牌，而在每个请求通过 grant_type=client_credentials 进行身份验证，而不是使用 refresh_token 作为授权类型。

因此，红帽单点登录支持在 OAuth2 Revocation 端点中撤销访问令牌。因此，如果需要，允许客户端撤销单独的访问令牌。

为了向后兼容，您可以继续使用之前的版本的行为。采用这种方法时，刷新令牌在成功与客户端凭证授予身份验证后，也会被创建用户会话。对于特定客户端，您可以在管理控制台中启用之前的行为，如下所示：

流程

1. 点菜单中的 Clients。
2. 点击您要修改的客户端。
3. 展开 OpenID Connect 兼容性模式 部分。
4. 将 Use Refresh Tokens For Client CredentialsGrant 切换到 ON。
5. 点 Save。

2.2.9. 移除了非标准令牌内省端点

在以前的版本中，Red Hat Single Sign-On 会公告两个内省端点： token_introspection_endpoint 和 introspection_endpoint。后者是由 RFC-8414 定义的。前者已弃用，现已被删除。

2.2.10. LDAP no-import 修复

在以前的 Red Hat Single Sign-On 版本中，当使用 Import Users OFF 配置 LDAP 供应商时，也可以更新用户，即使有些非 LDAP 映射的属性已更改。这种情况会导致行为混淆。属性显示为更新，但并没有更新。

例如，如果您试图使用 admin REST API 更新用户，并且用户有一些不正确的属性更改，则更新可能会发生。对于当前版本，无法更新，因此会立即通知您。

2.3. RH SSO 7.4

以下更改已从 Red Hat Single Sign-On 7.3 变为 Red Hat Single Sign-On 7.4。

2.3.1. 升级到 EAP 7.3

Red Hat Single Sign-On 服务器已升级为使用 EAP 7.3 作为底层容器。这个更改不会直接涉及任何特定的红帽单点登录服务器功能，但与迁移相关的一些变化。

2.3.1.1. 依赖项更新

依赖项更新至 EAP 7.3 服务器使用的版本。例如，Infinispan 组件版本现在是 9.3.1.Final。

2.3.1.2. 配置更改

standalone（-ha.xml 和 domain.xml 文件中有一些配置更改。按照升级 Red Hat Single Sign-On 服务器部分进行操作，以自动处理配置文件的迁移。

2.3.1.3. 跨数据中心复制更改

您需要将 RHDG 升级到 7.3 版本。旧版本可能仍然可以正常工作，但未经测试，因此无法保证它正常工作。

2.3.2. 身份验证流程更改

我们对身份验证流程进行了一些重构并改进，在迁移过程中需要注意它们。

2.3.2.1. REQUIRED 和 ALTERNATIVE 执行不支持相同的身份验证流

在以前的版本中，同一级别的身份验证流中可能会执行 REQUIRED 和 ALTERNATIVE 执行。这种方法有一些问题，我们在身份验证 SPI 中进行了重构，这意味着这已不再有效。如果在同一级别上配置了 ALTERNATIVE 和 REQUIRED 执行，则 ALTERNATIVE 执行被视为禁用。

因此，在迁移到这个版本时，现有的身份验证流程将被迁移，但会保留上一个版本的行为。如果身份

验证流程包含与 REQUIRED 执行相同的级别的 ALTERNATIVE 执行，ALTERNATIVE 执行会添加到单独的 REQUIRED 子流中。

此策略应确保与上一版本相同的或类似身份验证流程的行为。但是，您可以查看身份验证流程的配置，并重复检查它是否按预期工作。本建议适用于自定义验证程序实施的自定义身份验证流程。

2.3.2.2. 其它执行要求已删除

关于迁移，最重要的变化是消除对执行身份验证中的选项要求的支持，并使用 CONDITIONAL 要求替换它，这会增加灵活性。

在之前的版本中配置的选项验证器将被替换为 CONDITIONAL 子流。这些子流具有 Conditions - User Configured 条件配置为第一个执行，前面的 OPTIONAL authenticator（如 OTP Form）作为第二个执行。对于用户，身份验证过程中的行为与之前版本的行为匹配。

2.3.2.3. SPI 更改

Java 身份验证 SPI 和 Credential Provider SPI 中存在一些更改。

接口 Authenticator 没有改变，但如果您开发高级验证器来引入一些新凭证类型（凭证Model 的子类），则可能会受到影响。CredentialProvider 接口上存在更改，并引入一些新接口，如 CredentialValidator。

另外，如果您的验证器支持 OPTIONAL 执行要求，则可能会受到影响。建议您在服务器开发指南中仔细检查最新的身份验证示例以了解更多详细信息。

2.3.2.4. Freemarker 模板更改

自由标记模板中存在更改。如果您自己拥有适用于登录表单的自定义自由标记模板或一些帐户表单，则可能会会受到影响，特别是与 OTP 相关的表单。我们建议您查看此版本的自由标记模板中的更改，并根据模板进行调整。

2.3.3. 重复的顶层组

此发行版本解决了在域中创建重复的顶层组的问题。存在之前重复的组会使升级过程失败。如果使用 H2、MariaDB、MySQL 或 PostgreSQL 数据库，则 Red Hat Single Sign-On 服务器会受到此问题的影响。在启动升级前，检查服务器是否包含重复的顶层组。例如，可在数据库级别执行以下 SQL 查询以列出它们：

```
SELECT REALM_ID, NAME, COUNT(*) FROM KEYCLOAK_GROUP WHERE PARENT_GROUP is
NULL GROUP BY REALM_ID, NAME HAVING COUNT(*) > 1;
```

每个域中只能有一个顶层组名称。在升级前，应检查和删除重复项。升级中的错误包含消息 Change Set META-INF/jpa-changelog-9.0.1.xml::9.0.1- KEYCLOAK-12579-add-not-null-constraint::keycloak 失败。

2.3.4. 用户凭证更改

我们为存储用户凭证提供了更多灵活性。另外，每个用户都可以有多个相同类型的凭证，如多个 OTP 凭证。与数据库架构中存在一些变化，但上一版本中的凭证会更新为新格式。用户仍可以使用之前版本中定义的密码或 OTP 凭证登录。

2.3.5. 新的可选客户端范围

我们添加了 microprofile-jwt 可选客户端范围来处理 MicroProfile/JWT Auth 规范中定义的声明。这个新客户端范围定义了 protocol 映射程序，将经过身份验证的用户的用户名设置为 upn 声明，并将 realm 角色设置为组声明。

2.3.6. 改进了用户区域设置的处理

现在，如何选择登录页面的区域以及用户更新了区域设置时，进行了一些改进。详情请查看 服务器管理指南。

2.3.7. JavaScript 适配器中的旧承诺

您不再需要在 JavaScript 适配器中设置 promiseType，且两者同时可用。建议尽可能立即更新应用程序以使用原生保证 API（then 和 catch），因为旧 API（成功和错误）将在某个点上被删除。

2.3.8. 在服务器上部署脚本

到目前为止，管理员可以通过红帽单点登录管理控制台和 RESTful 管理 API 将脚本上传到服务器。这个功能现已被禁用。用户应该直接将脚本部署到服务器。如需更多详细信息，请查看 JavaScript Providers。

2.3.9. JavaScript 适配器中的客户端凭证

在以前的版本中，开发人员可以为 JavaScript 适配器提供客户端凭证。现在，这个功能已被移除，因为客户端的应用程序无法安全保存 secret。能够将 prompt=none 传播到默认的 IDP

我们已在名为 Accepts prompt=none forward from client 的 OIDC 身份提供程序配置中添加了一个切换，以标识可以处理包含 prompt=none 查询参数的 IDP。

在现在之前，在收到带有 prompt=none 的 auth 请求时，如果用户没有通过 IDP 进行身份验证，则域将返回 login_required 错误。现在，如果可以从现在确定默认 IDP（可以使用 kc_idp_hint 查询 param，或者为域设置默认 IDP）以及是否已为 IDP 启用了 Accepts prompt=none，则验证请求将转发到 IDP，检查用户是否已通过身份验证。

务必要注意，只有在指定了默认 IDP 时，这个切换才被考虑。在这种情况下，我们知道在不需要提示用户选择 IDP 的情况下转发 auth 请求。如果无法确定默认的 IDP，我们不能假设将使用哪个 IDP 来实现 auth 请求，因此不会执行请求转发。

2.3.10. 新的默认主机名供应商

请求和固定主机名供应商已被新的默认主机名供应商替代。请求和固定主机名供应商现已弃用，我们建议您尽快切换到默认主机名供应商。

2.3.11. 弃用或删除的功能

某些功能在状态有变化。

2.3.11.1. 令牌表示 Java 类中的已弃用方法

在 2038 年，Lint 无法再保存自 1970 年以来的秒数，因此我们正在努力将这些值更新到长值。令牌声明中还有更多问题。int 默认的结果是 JSON 表示的 0，而不应包含它。

有关已弃用和替换方法的具体方法的详情，请参阅 JavaDocs 文档。

2.3.11.2. 上传脚本

通过管理员其他端点/控制台上传脚本已弃用。它将在以后的发行版本中被删除。

2.3.12. 授权服务 Drools 策略

Authorization Services Drools Policy 已被删除。

2.4. RH-SSO 7.3

以下更改已从 RH-SSO 7.2 变为 RH-SSO 7.3。

2.4.1. 授权服务更改

我们添加了对 UMA 2.0 的支持。此版本的 UMA 规范中引入了一些重要的更改，它是如何从服务器获得权限的方式。

以下是 UMA 2.0 支持的主要变化。详情请参阅授权服务指南。

删除了授权 API

在 UMA 2.0(UMA 1.0)之前，客户端应用程序使用 Authorization API 从服务器获取权限，格式为 RPT。新版本的 UMA 规格已删除了从 Red Hat Single Sign-On 中删除的 Authorization API。在 UMA 2.0 中，现在可使用特定授权类型从令牌端点获取 RPT。详情请参阅授权服务指南。

删除授权 API

随着 UMA 2.0 的推出，我们决定利用令牌端点和 UMA 授权类型，以允许从红帽单点登录中获取 RPT，并避免具有不同 API。Entitlement API 提供的功能相同，仍然有可能获取一个或多个资源和范围的权限，如果没有提供资源或范围，则仍有可能获得一个或多个资源和范围的权限。详情请参阅授权服务指南。

UMA 发现端点的更改

UMA Discovery 文档已更改，请参阅授权服务指南。

Red Hat Single Sign-On Authorization JavaScript adapter 的更改

Red Hat Single Sign-On Authorization JavaScript adapter(keycloak-authz.js)已更改，以符合 UMA 2.0 引入的变化，同时保持之前的行为相同。主要变化在于如何调用 授权 和授权 方法，它们现在期望一个代表授权请求的特定对象类型。这个新对象类型提供了更多灵活性，它通过支持 UMA 授权类型支持从服务器获取权限的方式获得权限。详情请参阅授权服务指南。

> One of the main changes introduced by this release is that you are no longer required to exchange access tokens with RPTs in
> order to access resources protected by a resource server (when not using UMA).
> Depending on how the policy enforcer is configured on the resource server side, you can just send regular
> access tokens as a bearer token and permissions will still be enforced.

Red Hat Single Sign-On Authorization Client Java API 的更改

当升级到 Red Hat Single Sign-On Authorization Client Java API 的新版本时，您会注意到某

些表示类被移到 org.keycloak:keycloak:keycloak-core 的不同软件包中。

2.4.2. 客户端模板更改为客户端范围

添加了对客户端范围的支持，这需要在迁移期间进行一些关注。

客户端模板更改为客户端范围

客户端模板已更改为客户端范围。如果您有任何客户端模板，则会保留其协议映射程序和角色范围映射。

空格替换为名称

使用名称中的空格替换空格来重命名了名称中的客户端模板，因为客户端范围名称中不允许有空格。例如，我的 模板将更改为客户端范围 my_template。

将客户端范围链接到客户端

对于具有客户端模板的客户端，对应的客户端范围现在被添加为 客户端的默认客户端范围。因此，客户端上会保留协议映射程序和角色范围映射。

与现有客户端没有关联的域默认客户端范围

在迁移过程中，内置客户端范围列表添加到每个域，以及 Realm Default Client Scopes 列表。但是，现有客户端不会被升级，新的客户端范围不会被自动添加到它们。另外，所有协议映射程序和角色范围映射都保存在现有客户端上。在新版本中，当您创建新客户端时，它会自动附加 Realm Default Client Scopes，它没有附加任何协议映射程序。我们在迁移过程中不会更改现有客户端，因为无法正确地检测到自定义，例如，客户端具有协议映射程序。如果要更新现有客户端（删除协议映射程序），并使用客户端范围链接它们，您需要手动操作。

需要再次确认同意

客户端范围更改需要重构为同意。现在同意会指向客户端范围，而不是角色或协议映射程序。由于这一更改，之前确认用户的永久同意已经有效，用户需要在迁移后再次确认同意页面。

一些配置切换已删除

从角色详情中删除了交换机 范围范围（必需 ）。交换机 Consent Required 和 Consent 文本 已从协议映射程序详情中删除。这些交换机由 Client Scope 功能替代。

2.4.3. 新的默认客户端范围

我们已添加新的域默认客户端范围 角色和 web-origins。这些客户端范围包含协议映射程序，用于将用户的角色和允许 Web 来源添加到令牌中。在迁移过程中，这些客户端范围应自动添加到所有 OpenID Connect 客户端中作为默认客户端范围。因此，在数据库迁移完成后不需要设置。

2.4.3.1. 协议映射程序 SPI 添加了

与这一点相关，在（不支持）协议映射程序中会有一个小的补充。只有在您实施了自定义 ProtocolMapper 时，才能受到影响。ProtocolMapper 接口中有一个新的 getPriority（） 方法。该方法将默认实施设置为返回 0。如果您的协议映射程序实施依赖于访问令牌 realmAccess 或 resourceAccess 属性中的角色，您可能需要增加映射程序的优先级。

2.4.3.2. 受众解决

所有经过身份验证的用户在令牌中有至少一个客户端角色的客户端现在会自动添加到访问令牌中的声明中。另一方面，访问令牌不会自动包含发布该前端客户端的读者。详情请查看 服务器管理指南。

2.4.4. 升级到 EAP 7.2

Red Hat Single Sign-On 服务器已升级为使用 EAP 7.2 作为底层容器。这不会直接涉及任何特定的 Red Hat Single Sign-On 服务器功能，但有一些与迁移相关的更改，值得提到。

依赖项更新

依赖项已更新至 EAP 7.2 服务器使用的版本。例如，Inftables 现在是 9.3.1.Final。

配置更改

standalone（-ha.xml 和 domain.xml 文件中有一些配置更改。您应该遵循 第 3.1.2 节 “升级 Red Hat Single Sign-On 服务器” 部分自动处理配置文件迁移。

跨数据中心复制更改

- 您需要将 RHDG 服务器升级到 7.3 版本。旧版本可能仍然可以正常工作，但无法保证，我们再无法对其进行测试。
- 需要使用在 Red Hat Single Sign- On 配置中将 protocolVersion 属性添加到 remote-store 元素的配置中。需要这个功能，因为需要降级 HotRod 协议的版本与 RHDG 7.3 使用的版本兼容。

2.4.5. 主机名配置

在以前的版本中，建议使用过滤器来指定允许的主机名。现在可以设置固定主机名，这有助于确保使用有效主机名并允许内部应用程序通过替代 URL 调用 Red Hat Single Sign-On，例如内部 IP 地址。建议您在生产环境中切换到这个方法。

2.4.6. JavaScript 适配器承诺

要将原生 JavaScript 适配器用于 JavaScript 适配器，需要在 init 选项中将 promiseType 设置为 native。

过去，如果本地承诺提供的打包程序已被返回，为旧的 Keycloak 承诺和原生承诺提供。这是因为错误处理器在原生错误事件之前没有始终设置问题，从而导致 Uncaught（承诺） 错误。

2.4.7. Microsoft Identity Provider 更新为使用 Microsoft Graph API

Red Hat Single Sign-On 中的 Microsoft Identity Provider 实施，用于依赖 Live SDK 端点来获取授权并获取用户配置集。自 2018 年 11 月起，Microsoft 正移除对 Live SDK API 的支持，而是使用新的 Microsoft Graph API。Red Hat Single Sign-On 身份提供程序已更新为使用新的端点，因此如果此集成正在使用，请确保升级到最新的 Red Hat Single Sign-On 版本。

在 "Live SDK 应用程序" 下注册的传统客户端应用程序不会因为应用程序的 id 格式更改而用于 Microsoft Graph 端点。如果遇到错误，表示 目录中未找到应用程序标识符，则必须在 Microsoft Application Registration Portal 中再次注册客户端应用程序以获取新的应用程序 ID。

2.4.8. Google Identity Provider 更新为使用 Google Sign-in 身份验证系统

红帽单点登录中的 Google 身份提供程序实施，用于依赖 Google+ API 端点来授权并获取用户配置集。从 2019 年 3 月开始，Google 正在移除对 Google+ API 的支持，而是使用新的 Google Sign-in 身份验证系统。Red Hat Single Sign-On 身份提供程序已更新为使用新的端点，因此如果此集成正在使用，请确保升级到最新的 Red Hat Single Sign-On 版本。

如果遇到错误，表示 目录中找不到应用程序标识符，则必须在 Google API 控制台 门户中再次注册客户端应用程序，以获取新的应用程序 ID 和 secret。

对于 Google+ 用户信息端点提供的非标准声明，您可能需要调整自定义映射程序，并由 Google Sign-in API 提供。有关可用声明的最新信息，请参阅 Google 文档。

2.4.9. LinkedIn social Broker 更新至 LinkedIn API 版本 2

使用 LinkedIn 相应地，所有开发人员都需要迁移到其 API 和 OAuth 2.0 版本 2.0。因此，我们更新了 LinkedIn social Broker。

使用这个代理的现有部署可能会在使用 LinkedIn API 版本 2 时开始遇到错误。这个错误可能与没有为

客户端应用程序授予权限（在验证过程中无法授权访问 Profile API 或请求特定的 OAuth2 范围）相关。

即使对于新创建的 LinkedIn 客户端应用程序，您需要确保客户端能够请求 r_liteprofile 和 r_emailaddress OAuth2 范围，以及客户端应用程序可以从 https://api.linkedin.com/v2/me 端点获取当前成员的配置集。

由于 LinkedIn 对这些隐私限制导致对成员的信息的访问以及当前成员的 Profile API 返回的一组有限声明集，LinkedInSocial Broker 现在使用成员的电子邮件地址作为默认用户名。这意味着，在身份验证过程中发送授权请求时，r_emailaddress 会始终设置。

2.5. RH-SSO 7.2

以下更改已从 RH-SSO 7.1 升级到 RH-SSO 7.2。

2.5.1. 新的密码哈希算法

我们添加了两个新的密码哈希算法（pbkdf2-sha256 和 pbkdf2-sha512）。新域将使用 pbkdf2-sha256 哈希算法，并带有 27500 哈希迭代。由于 pbkdf2-sha256 比 pbkdf2 快于 pbkdf2，因此迭代从 20000 增加到 27500。

如果密码策略包含哈希算法（未指定）和迭代(20000)的默认值，则会升级现有 realm。如果您更改了散列迭代，如果您需要使用更安全的哈希算法，则需要手动更改为 pbkdf2-sha256。

2.5.2. ID 令牌需要 scope=openid

在 RH-SSO 7.0 中，无论在授权请求中存在 scope=openid 查询参数，都会返回 ID 令牌。这会根据 OpenID Connect 规格不正确。

在 RH-SSO 7.1 中，我们向适配器添加了此查询参数，但保留了旧行为以容纳迁移。

在 RH-SSO 7.2 中，此行为已更改，现在需要使用 scope=openid 查询参数来将请求标记为 OpenID Connect 请求。如果省略了此查询参数，则不会生成 ID Token。

2.5.3. Microsoft SQL Server 需要额外的依赖项

Microsoft JDBC Driver 6.0 需要额外的依赖项添加到 JDBC 驱动程序模块。如果您使用 Microsoft

SQL Server 时观察了 NoClassDefFoundError 错误，请在 JDBC 驱动程序 module.xml 文件中添加以下依赖项：

```
<module name="javax.xml.bind.api"/>
```

2.5.4. 在 OpenID Connect 身份验证响应中添加了 session_state 参数

OpenID Connect Session Management 规格要求参数 session_state 存在于 OpenID Connect 身份验证响应中。

在 RH-SSO 7.1 中，我们没有此参数，但现在 Red Hat Single Sign-On 会按照规范的要求添加此参数。

但是，一些 OpenID Connect / OAuth2 适配器，特别是旧的 Red Hat Single Sign-On 适配器（如 RH-SSO 7.1 和更早的版本）可能会遇到这个新参数的问题。

例如，在成功对客户端应用进行身份验证后，参数始终存在于浏览器 URL 中。如果您使用 RH-SSO 7.1 或旧的 OAuth2 / OpenID Connect 适配器，则禁用将 session_state 参数添加到身份验证响应中可能很有用。这可以在 Red Hat Single Sign-On 管理控制台中为特定的客户端完成，在带有 OpenID Connect Compatibility Modes 部分的客户端详情中，如 第 4.1 节 “与旧的适配器兼容” 所述。还有 Exclude Session State From Authentication Response 开关，它可以被打开，以防止将 session_state 参数添加到身份验证响应中。

2.5.5. Microsoft Identity Provider 更新为使用 Microsoft Graph API

Red Hat Single Sign-On up to version 7.2.4 中的 Microsoft Identity Provider 实施依赖于 Live SDK 端点来获取用户配置集。自 2018 年 11 月起，Microsoft 正移除对 Live SDK API 的支持，而是使用新的 Microsoft Graph API。Red Hat Single Sign-On 身份提供程序已更新为使用新的端点，因此如果此集成正在使用，请确保升级到 Red Hat Single Sign-On 版本 7.2.5 或更高版本。

在 "Live SDK 应用程序" 下注册的传统客户端应用程序不会因为应用程序的 id 格式更改而用于 Microsoft Graph 端点。如果遇到错误，表示 目录中未找到应用程序标识符，则必须在 Microsoft Application Registration Portal 中再次注册客户端应用程序以获取新的应用程序 ID。

2.5.6. Google Identity Provider 更新为使用 Google Sign-in 身份验证系统

Red Hat Single Sign-On up to version 7.2.5 中的 Google Identity Provider 实现依赖于 Google+ API 端点来授权并获取用户配置集。从 2019 年 3 月开始，Google 正在移除对 Google+ API 的支持，而

是使用新的 Google Sign-in 身份验证系统。Red Hat Single Sign-On 身份提供程序已更新为使用新的端点，因此如果此集成正在使用，请确保升级到 Red Hat Single Sign-On 版本 7.2.6 或更高版本。

如果遇到错误，表示 目录中找不到应用程序标识符，则必须在 Google API 控制台 门户中再次注册客户端应用程序，以获取新的应用程序 ID 和 secret。

对于 Google+ 用户信息端点提供的非标准声明，您可能需要调整自定义映射程序，并由 Google Sign-in API 提供。有关可用声明的最新信息，请参阅 Google 文档。

2.5.7. LinkedIn social Broker 更新至 LinkedIn API 版本 2

使用 LinkedIn 相应地，所有开发人员都需要迁移到其 API 和 OAuth 2.0 版本 2.0。因此，我们更新了 LinkedIn Social Broker，因此如果此集成正在使用，请确保升级到 Red Hat Single Sign-On 7.2.6 或更高版本。

使用这个代理的现有部署可能会在使用 LinkedIn API 版本 2 时开始遇到错误。这个错误可能与没有为客户端应用程序授予权限（在验证过程中无法授权访问 Profile API 或请求特定的 OAuth2 范围）相关。

即使对于新创建的 LinkedIn 客户端应用程序，您需要确保客户端能够请求 r_liteprofile 和 r_emailaddress OAuth2 范围，以及客户端应用程序可以从 https://api.linkedin.com/v2/me 端点获取当前成员的配置集。

由于 LinkedIn 对这些隐私限制导致对成员的信息的访问以及当前成员的 Profile API 返回的一组有限声明集，LinkedInSocial Broker 现在使用成员的电子邮件地址作为默认用户名。这意味着，在身份验证过程中发送授权请求时，r_emailaddress 会始终设置。

2.6. RH-SSO 7.1

以下更改已从 RH-SSO 7.0 到 RH-SSO 7.1。

2.6.1. realm 密钥

对于 RH-SSO 7.0，只有一组键可以与一个域关联。这意味着，在更改密钥时，所有当前的 cookie 和令牌都会无效，所有用户都必须重新验证。对于 RH-SSO 7.1，增加了对一个域的多个密钥的支持。在任何给定时间，一个密钥集就是用于创建签名的有效集，但可使用多个密钥来验证签名。这意味着可以验证旧的 cookie 和令牌，然后使用新的签名刷新，从而允许用户在更改密钥时保持身份验证。另外，还有一些更改如何通过 Admin Console 和 Admin REST API 管理密钥；有关更多详情，请参阅《服务器管理指南》中的 Realm Keys。

要允许无缝密钥轮转，您必须从客户端适配器中删除硬编码密钥。只要未指定域密钥，客户端适配器将自动从服务器检索密钥。客户端适配器还会在密钥轮转时自动检索新密钥。

2.6.2. 客户端重定向 URI 匹配

对于 RH-SSO 7.0，在与客户端的有效重定向 URI 匹配时，查询参数将被忽略。对于 RH-SSO 7.1，查询参数不再忽略。如果您需要在重定向 URI 中包含查询参数，您必须在客户端的有效重定向 URI 中指定查询参数（例如 https://hostname/app/login?foo=bar）或使用通配符（例如 https://hostname/app/login/*）。在有效的重定向 URI 中也不再允许片段（即 https://hostname/app#fragment）。

2.6.3. 自动重定向到身份提供程序

对于 RH-SSO 7.1，身份提供程序无法设置为默认的身份验证提供程序。要自动重定向到 RH-SSO 7.1 的身份提供商，现在您必须配置身份提供程序重定向器。如需更多信息，请参阅 *《服务器管理指南》* 中的默认身份提供商。如果您之前设置了默认的身份验证提供程序选项，则当服务器升级到 RH-SSO 7.1 时，这个值会自动用作身份提供程序重定向的值。

2.6.4. 管理员 REST API

对于 RH-SSO 7.0，如果未指定 maxResults 查询参数，则 Admin REST API 中的分页端点会返回所有结果。当返回大量结果（例如，用户）时，这可能会导致临时高负载和请求超时的问题。对于 RH-SSO 7.1，如果没有指定 maxResults，则返回最多 100 个结果。您可以通过将 maxResults 指定为 -1 来返回所有结果。

2.6.5. 服务器配置

对于 RH-SSO 7.0，服务器配置在 keycloak-server.json 文件和 standalone/domain.xml 或 domain.xml 文件之间分割。对于 RH-SSO 7.1，keycloak-server.json 文件已被删除，所有服务器配置都通过 standalone.xml 或 domain.xml 文件完成。RH-SSO 7.1 的升级流程会自动将服务器配置从 keycloak-server.json 文件迁移到 standalone.xml 或 domain.xml 文件。

2.6.6. SAML 断言中的密钥加密算法

对于 RH-SSO 7.1，SAML 断言中的密钥和文档现在使用 RSA-OAEP 加密方案进行加密。要使用加密的断言，请确保您的服务供应商支持这个加密方案。如果您的服务提供商不支持 RSA-OAEP，则可将 RH-SSO 配置为使用旧的 RSA-v1.5 加密方案，方法是使用系统属性"keycloak.saml.key_trans.rsa_v1.5" 的服务器。如果您这样做，应该尽快升级您的服务提供商，以便可以恢复到更安全的 RSA-OAEP 加密方案。

第 3 章 升级 RED HAT SINGLE SIGN-ON 服务器

Red Hat Single Sign-On 服务器的升级或迁移流程取决于软件以前的版本。

- 如果您要升级到一个新的次版本，例如从 7.5.x 升级到 7.6，请按照 Minor Upgrades 中的步骤操作。
- 如果您要从 Keycloak 18.0.0 迁移，请按照 Minor Upgrades 中的步骤操作。
- 如果您要升级到新的微版本，例如从 7.5.2 升级到 7.5.3，请按照 Micro Upgrades 中的步骤操作。

3.1. 执行次要升级

3.1.1. 准备升级

在升级前，请注意您需要执行升级步骤的顺序。特别是，在升级适配器前，请务必升级 Red Hat Single Sign-On 服务器。

警告

在 Red Hat Single Sign-On 的次要升级中，所有用户会话都会丢失。升级后，所有用户都需要再次登录。

流程

1. 备份旧安装（配置、主题等）。
2. 使用关系数据库文档中的说明备份数据库。
3. 升级红帽单点登录服务器。

升级后，该数据库将不再与旧服务器兼容。

4. 如果您需要恢复升级，首先恢复旧的安装，然后从备份副本中恢复数据库。

5. 升级适配器。

3.1.2. 升级 Red Hat Single Sign-On 服务器

按照以下步骤确保服务器升级成功：

- 先在非生产环境中测试升级，以防止生产环境中的出现任何安装问题，
- 在升级适配器前，先升级 Red Hat Single Sign-On 服务器。另外，在升级适配器前，还确保升级的服务器可以在生产环境中正常工作。

警告

因为特定于您的安装，这个升级步骤可能需要修改。有关可能影响升级的手动更改的详情，请参阅 特定于发布的更改。

根据您用于安装的方法，从 ZIP 文件或 RPM 升级服务器。

3.1.2.1. 从 ZIP 文件升级服务器

先决条件

- 处理任何开放事务并删除 data/tx-object-store/ 事务目录。

流程

1. 下载新服务器存档。

2. 将下载的存档移动到所需位置。

3. 提取存档。这一步会安装最新 Red Hat Single Sign-On 版本的一个干净实例。

4. 对于单机安装，请将上一安装中的 RHSSO_HOME/standalone/ 目录复制到新安装中的 目录。

   对于域安装，请将之前安装的 RHSSO_HOME/domain/ 目录复制到新安装中的 目录。

   对于域安装，请创建空目录 RHSSO_HOME/domain/deployments。

   注意：bin 目录中的文件不应由之前版本中的文件覆盖。更改应手动进行。

5. 复制添加到模块目录中的任何自定义模块。

6. 继续操作，运行服务器升级脚本。

3.1.2.2. 从 RPM 升级服务器

先决条件

- 处理任何开放事务并删除 /var/opt/rh-sso7/lib/keycloak/standalone/data/tx-object-store/ transaction 目录。

流程

1. 订阅正确的包含 Red Hat Single Sign-On 的软件仓库。

   Red Hat Enterprise Linux 7：

   ```
   subscription-manager repos --enable=rh-sso-7.6-for-rhel-7-x86_64-rpms
   ```

Red Hat Enterprise Linux 8：

```
subscription-manager repos --enable=rh-sso-7.6-for-rhel-8-x86_64-rpms
```

2. 为 Red Hat Single Sign-On 禁用旧的产品存储库：

```
subscription-manager repos --disable=rh-sso-7.5-for-rhel-8-x86_64-rpms
```

3. 检查软件仓库列表：

```
dnf repolist
```

```
Updating Subscription Management repositories.
repo id repo name
rh-sso-7.6-for-rhel-8-x86_64-rpms Single Sign-On 7.6 for RHEL 8 x86_64 (RPMs)
rhel-8-for-x86_64-appstream-rpms Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
rhel-8-for-x86_64-baseos-rpms Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)
```

4. 备份所有修改后的配置文件和自定义模块。

5. 使用 dnf upgrade 升级到新的 Red Hat Single Sign-On 版本。

RPM 升级过程不会替换任何修改后的配置文件。相反，此过程会为新红帽单点登录版本的默认配置创建 .rpmnew 文件。

6. 要激活新版本中的任何新功能，如新子系统，手动将每个 .rpmnew 文件合并到您现有的配置文件中。

7. 复制添加到模块目录中的任何自定义模块。

8. 继续操作，运行服务器升级脚本。

注意

Red Hat Single Sign-On RPM 服务器发行版本使用

RHSSO_HOME=/opt/rh/rh-sso7/root/usr/share/keycloak

在调用以下迁移脚本时使用它。

3.1.3. 运行服务器升级脚本

根据您的安装，运行适用于您的情况的适当升级脚本：

- 独立模式
- 独立高可用性模式
- 域模式
- 域集群模式

3.1.3.1. 运行独立模式升级脚本

流程

1. 如果您使用不同于默认值的配置文件，请编辑迁移脚本以指定新文件名。
2. 停止服务器。
3. 运行升级脚本：

```
bin/jboss-cli.sh --file=bin/migrate-standalone.cli
```

3.1.3.2. 运行 standalone-High Availability Mode 升级脚本

对于独立高可用性(HA)模式，所有实例必须同时升级。

流程

1. 如果您使用不同于默认值的配置文件，请编辑迁移脚本以指定新文件名。
2. 停止服务器。
3. 运行升级脚本：

```
bin/jboss-cli.sh --file=bin/migrate-standalone-ha.cli
```

3.1.3.3. 运行 Domain Mode 升级脚本

对于域模式，所有实例必须同时升级。

流程

1. 如果您更改了配置集名称，您必须编辑升级脚本，以在脚本开始时更改变量。
2. 编辑域脚本，使其包含 keycloak-server.json 文件的位置。
3. 停止服务器。
4. 在域控制器上运行升级脚本

```
bin/jboss-cli.sh --file=bin/migrate-domain.cli
```

3.1.3.4. 运行 Domain-clustered 模式升级脚本

对于 domain-clustered 模式，所有实例必须同时升级。

流程

流程

1. 如果您更改了配置集名称，您必须编辑升级脚本，以在脚本开始时更改变量。

2. 编辑 domain-clustered 脚本，使其包含 keycloak-server.json 文件的位置。

3. 停止服务器。

4. 仅在域控制器上运行升级脚本：

```
bin/jboss-cli.sh --file=bin/migrate-domain-clustered.cli
```

3.1.4. 数据库迁移

Red Hat Single Sign-On 可以自动迁移数据库架构，也可以选择手动操作。默认情况下，当您首次启动新安装时，数据库会自动迁移。

3.1.4.1. 自动关系数据库迁移

要启用对数据库 schema 的自动升级，请将 migrationStrategy 属性值设置为 更新默认 connectionJpa 供应商：

```
<spi name="connectionsJpa">
  <provider name="default" enabled="true">
    <properties>
      ...
      <property name="migrationStrategy" value="update"/>
    </properties>
  </provider>
</spi>
```

或运行此 CLI 命令：

```
/subsystem=keycloak-server/spi=connectionsJpa/provider=default/:map-
put(name=properties,key=migrationStrategy,value=update)
```

使用此设置启动服务器时，如果新版本中更改了数据库，则会自动迁移数据库。

在具有数百万记录的大量表中创建索引，可轻松花费大量时间，并可能对升级造成主要服务中断。对于这些情况，我们添加了用于自动创建索引的阈值（记录数）。默认情况下，这个阈值是 300000 记录。当记录数量高于阈值时，索引不会自动创建，且服务器日志中会有一个警告消息，包括稍后可以手动应用的 SQL 命令。

要更改阈值，请设置 indexCreationThreshold 属性，默认 connection Liquibase 供应商的值：

```
<spi name="connectionsLiquibase">
  <provider name="default" enabled="true">
    <properties>
      <property name="indexCreationThreshold" value="300000"/>
    </properties>
  </provider>
</spi>
```

或运行此 CLI 命令：

```
/subsystem=keycloak-server/spi=connectionsLiquibase/:add(default-provider=default)
/subsystem=keycloak-server/spi=connectionsLiquibase/provider=default/:add(properties=
{indexCreationThreshold => "300000"},enabled=true)
```

3.1.4.2. 手动关系数据库迁移

要启用对数据库 schema 的手动升级，请将默认的 connectionJpa 供应商的 migrationStrategy 属性值设置为 manual ：

```
<spi name="connectionsJpa">
  <provider name="default" enabled="true">
    <properties>
      ...
      <property name="migrationStrategy" value="manual"/>
    </properties>
  </provider>
</spi>
```

或运行此 CLI 命令：

```
/subsystem=keycloak-server/spi=connectionsJpa/provider=default/:map-
put(name=properties,key=migrationStrategy,value=manual)
```

使用此配置启动服务器时，它会检查是否需要迁移数据库。所需更改写入到 SQL 文件中，您可以检查并手动针对数据库运行。有关如何将此文件应用到数据库的详情，请查看您使用的关系数据库的文档。在更改写入到该文件后，服务器会退出。

3.1.5. Theme migration

如果您创建了任何自定义它们，则必须将其迁移到新服务器。对内置主题的任何更改可能需要根据您自定义哪些方面在自定义中反映在自定义中。

您必须将自定义它们从旧服务器复制到新服务器，将其复制到新服务器目录中。 之后，您需要查看以下更改并考虑更改是否需要应用于您的自定义主题。

概述：

- 如果您自定义了以下所有更改的模板，则需要将模板与基础主题进行比较，以查看您是否需要应用了更改。
- 如果您已自定义了任何一种方式，并且要扩展红帽单点登录，那么您需要审查对风格的更改。如果要扩展基础，您可以跳过这一步。
- 如果您有自定义消息，您可能需要更改密钥或值或添加其他消息。

每个步骤都会详细介绍更改列表的下方。

3.1.5.1. 主题更改 RH-SSO 7.3

模板

- account: account.ftl
- 帐户：application.ftl
- 账户： resource-detail.ftl（新的）
- account: resources.ftl（新的）

- **account: template.ftl**
- **account: totp.ftl**
- **email-html: email-test.ftl**
- **email-html: email-verification-with-code.ftl（新的）**
- **email-html: email-verification.ftl**
- **email-html: event-login_error.ftl**
- **email-html: event-removed_totp.ftl**
- **email-html: event-update_password.ftl**
- **email-html: event-update_totp.ftl**
- **email-html: executeActions.ftl**
- **email-html: identity-provider-link.ftl**
- **email-html: password-reset.ftl**
- **email-text：电子邮件验证-带有码.ftl（新的）**
- **email-text：电子邮件验证.ftl**

- email-text： executeActions.ftl
- Email-text: identity-provider-link.ftl
- email-text：password-reset.ftl
- 登录：cli_splash.ftl（新的）
- login: code.ftl
- login: error.ftl
- login: info.ftl
- 登录：login-config-totp-text.ftl（新的）
- login: login-config-totp.ftl
- Login: login-idp-link-confirm.ftl
- login: login-idp-link-email.ftl
- login: login-oauth-grant.ftl
- login: login-page-expired.ftl
- login: login-reset-password.ftl

- **login: login-totp.ftl**
- **login: login-update-password.ftl**
- **login: login-update-profile.ftl**
- **login: login-verify-email-code-text.ftl（新的）**
- **login: login-verify-email.ftl**
- **login: login-x509-info.ftl**
- **login: login.ftl**
- **login: register.ftl**
- **login: template.ftl**
- 登录名： **terms.ftl**
- 欢迎：**index.ftl（新的）**

消息

- **account: message_en.properties**
- **admin: admin-messages_en.properties**

- 电子邮件： **message_en.properties**
- **login: message_en.properties**

样式

- 登录：**login-rhsso.cs（新的）**
- 欢迎： **welcome-rhsso.css**

3.1.5.2. 主题会改变 RH-SSO 7.2

模板

- **account: account.ftl**
- 帐户：**application.ftl**
- 帐户： **federatedIdentity.ftl**
- 帐户：**password.ftl**
- **account： sessions.ftl**
- **account: template.ftl**
- **account: totp.ftl**
- **admin: index.ftl**

- 电子邮件： email-test.ftl（新的）
- 电子邮件：电子邮件验证.ftl
- 电子邮件： event-login_error.ftl
- 电子邮件： event-removed_totp.ftl
- 电子邮件： event-update_password.ftl
- 电子邮件： event-update_totp.ftl
- 电子邮件： executeActions.ftl
- 电子邮件： identity-provider-link.ftl
- 电子邮件：password-reset.ftl
- 登录名： bypass_kerberos.ftl（删除）
- login: error.ftl
- login: info.ftl
- login: login-config-totp.ftl
- login: login-idp-link-email.ftl

- login: login-oauth-grant.ftl
- 登录：login-page-expired.ftl（新的）
- login: login-reset-password.ftl
- login: login-totp.ftl
- login: login-update-password.ftl
- login: login-update-profile.ftl
- login: login-verify-email.ftl
- login: login-x509-info.ftl（新的）
- login: login.ftl（新的）
- 登录名： register.ftl（新的）
- login: template.ftl（新的）
- 登录名： terms.ftl（新的）

消息

- account: message_en.properties

- admin: admin-messages_en.properties
- admin: message_en.properties
- 电子邮件： message_en.properties
- login: message_en.properties

样式

- account: account.css
- login: login.css

3.1.5.3. 主题更改 RH-SSO 7.1

模板

- account: account.ftl
- 帐户： federatedIdentity.ftl
- account: totp.ftl
- login: info.ftl
- login: login-config-totp.ftl
- login: login-reset-password.ftl

- login: login.ftl

消息

- 帐户：编辑AccountHtmlTtile 重命名以编辑AccountHtmlTitle
- account: role_uma_authorization added
- login: loginTotpStep1 值已更改
- login: invalidPasswordGenericMessage added
- login: invlidRequesterMessage renamed to invalidRequesterMessage
- login: clientDisabledMessage added

样式

- account: account.css
- login: login.css

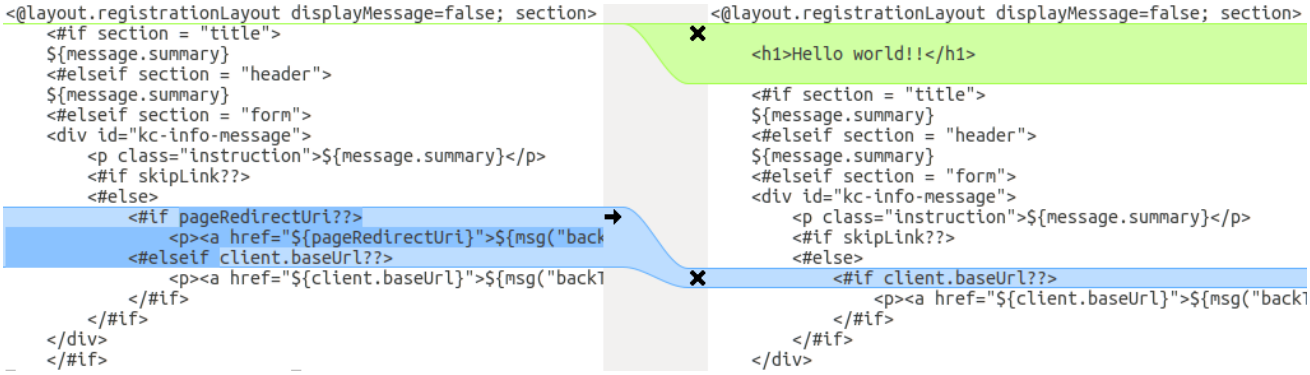
3.1.5.4. 迁移模板

如果您已自定义了任何模板，您需要仔细检查对模板所做的更改，以确定是否需要将这些更改应用到您的自定义模板。很可能需要对自定义模板应用相同的更改。如果您还没有自定义任何列出的模板，您可以跳过本节。

最佳实践是使用 diff 工具比较模板，以了解您可能需要对自定义模板进行的更改。如果您只进行了小更改，将更新的模板与自定义模板进行比较。但是，如果您进行了很多更改，将新模板与自定义旧模板进行比较，因为这将向您展示您需要做了哪些更改。

以下屏幕截图比较了 Login theme 和 example custom theme 中的 info.ftl 模板：

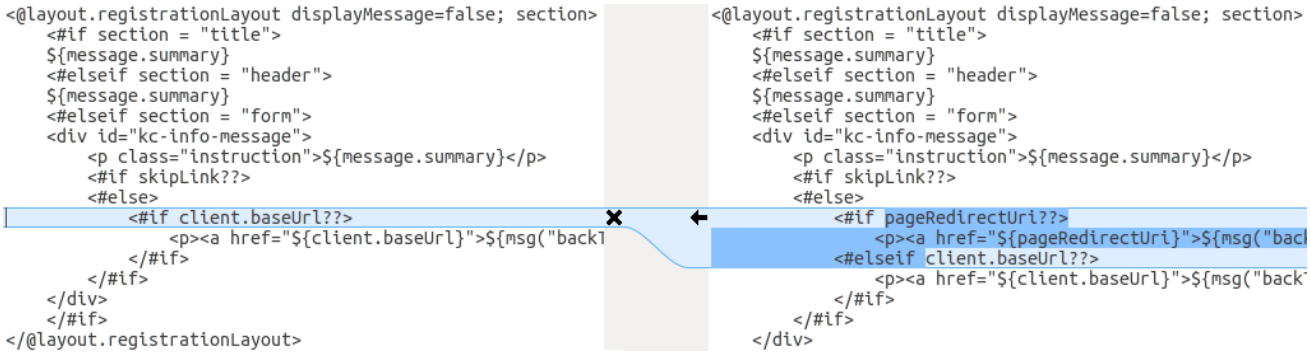
登录主题模板的更新版本与示例自定义登录主题模板进行比较



从这个比较上，很容易确认第一个更改(Hello world!!)是一个自定义，而第二次更改（如果页面 RedirectUri）是对主题的基本更改。通过将第二个更改复制到自定义模板，您可以成功更新自定义模板。

对于其他方法，以下屏幕截图将旧安装中的 info.ftl 模板与新安装中的更新的 info.ftl 模板进行比较：

旧安装中的登录主题模板与登录主题模板的更新版本进行比较



从此比较便于识别基本模板中的变化。然后，您必须手动对修改的模板进行相同的更改。由于此方法不像第一种方法那样简单，因此仅当第一个方法不可行时，才使用此方法。

3.1.5.5. 迁移消息

如果您添加了对其它语言的支持，则需要应用上述所有更改。如果您尚未添加对其他语言的支持，您可能不需要更改任何内容；如果您在主题中更改了受影响的消息，您只需要进行更改。

对于添加的值，请查看主题中消息的值，以确定您是否需要自定义该消息。

对于重命名的密钥，请重命名自定义主题中的密钥。

对于 changed 的值，检查基础主题中的值，以确定是否需要更改您的自定义主题。

3.1.5.6. 迁移风格

如果您要从键或 rh-sso 继承样式，您可能需要更新您的自定义风格，以反映内置它们对样式所做的更改。

最佳实践是使用 diff 工具比较旧服务器安装与新服务器安装之间样式表的更改。

例如，使用 diff 命令：

```
$ diff RHSSO_HOME_OLD/themes/keycloak/login/resources/css/login.css \
RHSSO_HOME_NEW/themes/keycloak/login/resources/css/login.css
```

检查更改，并确定它们是否影响您的自定义风格。

3.2. 执行微小升级

3.2.1. ZIP/installer 安装补丁

可以从红帽客户门户下载 适用于 RH-SSO 的 ZIP 安装的补丁。

对于受管域环境中的多个 RH-SSO 主机，单个主机可以从您的 RH-SSO 域控制器进行补丁。

除了应用补丁程序外，您还可以回滚补丁应用程序。

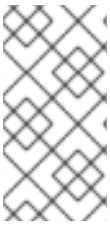
3.2.1.1. ZIP 安装修复的重要备注

- 如果您应用了一个更新模块的补丁程序，则在运行时使用的新 patched JAR 将存储在 RHSSO_HOME/modules/system/layers/base/.overlays/*PATCH_ID*/*MODULE* 中。原始未修补的文件保留在 RHSSO_HOME/modules/system/layers/base/*MODULE* 中，但这些 JAR 不会在运行时使用。

- 为了显著减少 RH-SSO 7 的累积补丁发行版本的大小，您无法对累积修补程序进行部分回滚。对于已应用的补丁，您将只能回滚整个补丁。

  例如，如果您将 CP03 应用到 RH-SSO 7.0.0，您将无法回滚到 CP01 或 CP02。如果您希望能够回滚到每个累积补丁版本，则必须按照发布顺序单独应用每个累积补丁。

3.2.1.2. 应用补丁



注意

使用 RPM 方法安装的 RH-SSO 服务器不能按照以下说明进行更新。请参阅应用补丁的 RPM。

您可以使用 管理 CLI 或管理控制台 将下载的补丁应用到 RH-SSO 服务器。

流程

1. 从红帽客户门户网站下载补丁文件，地址为 https://access.redhat.com/downloads/。

2. 在管理 CLI 中，使用以下命令应用补丁，包括到补丁文件的适当路径：

```
patch apply /path/to/downloaded-patch.zip
```

注意

要修补受管域中的另一个 RH-SSO 主机，您可以使用 --host= 参数指定 RH-SSO 主机名。例如：

```
patch apply /path/to/downloaded-patch.zip --host=my-host
```

如果尝试应用补丁时存在任何冲突，补丁工具将会发出警告。如果存在冲突，输入 patch --help 来使用可用的参数重新运行命令，并指定如何解决冲突。

3. 重启 RH-SSO 服务器，使补丁生效：

```
shutdown --restart=true
```

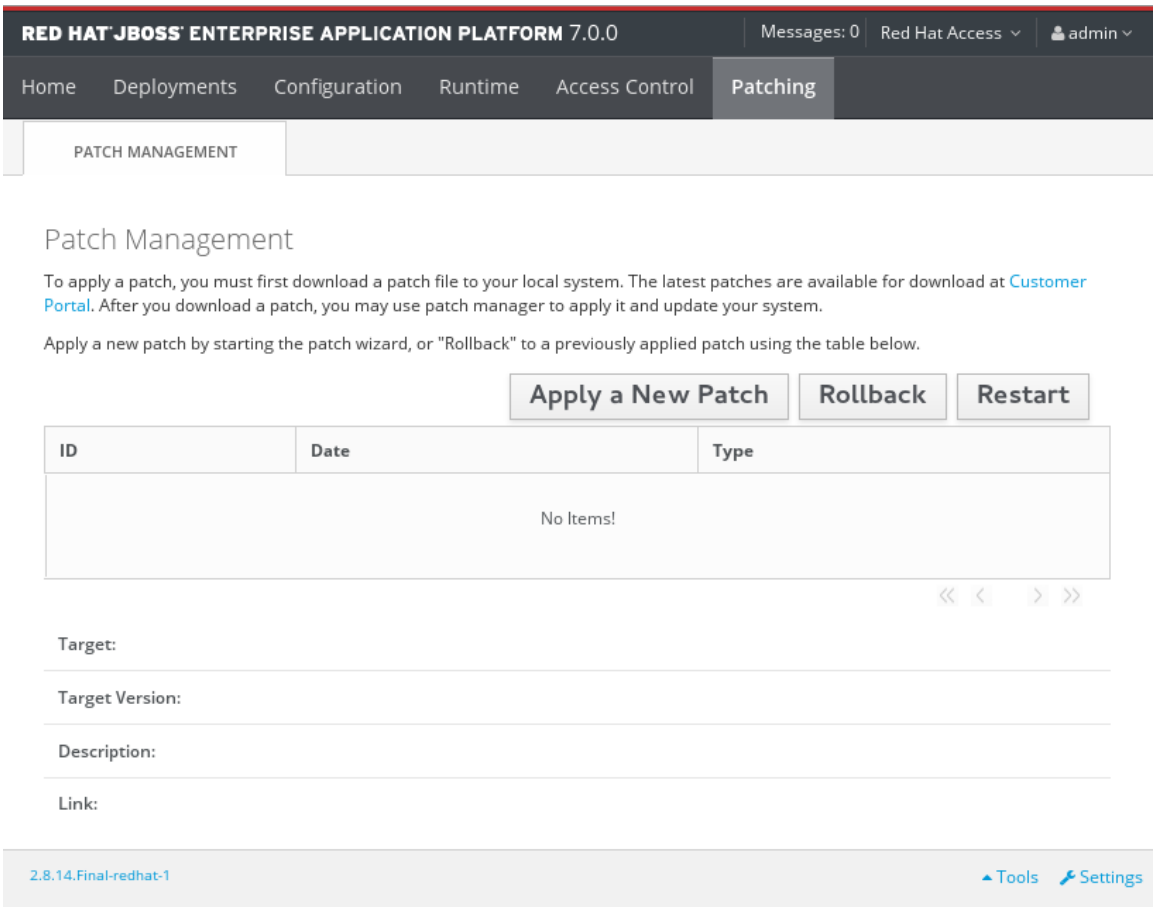
流程

1. 从红帽客户门户网站下载补丁文件，地址为 https://access.redhat.com/downloads/。

2. 打开 管理控制台，再导航到 Patch Management 视图。

   a. 对于单机服务器，请单击 补丁 选项卡。

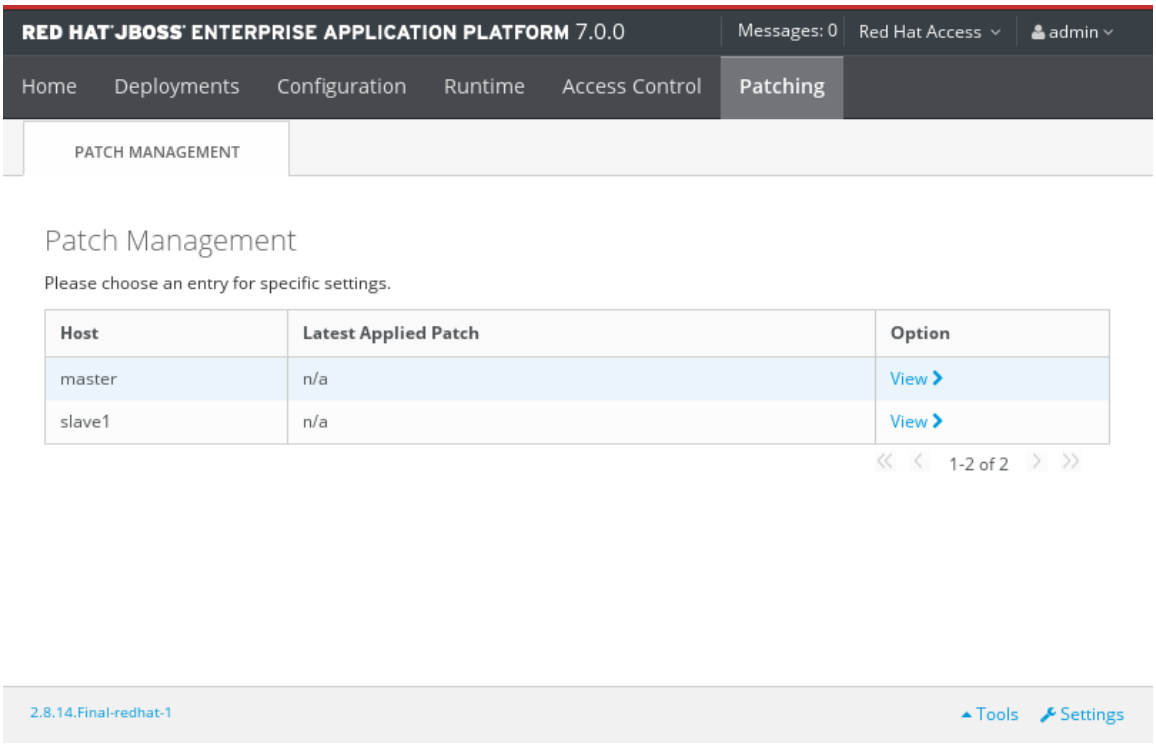
单机服务器的补丁管理屏幕



b.

对于受管域中的服务器，单击 **patching** 选项卡，然后选择要从表中修补的主机，然后单击 **View**。

受管集群的补丁管理屏幕



3.

点击 应用新补丁。

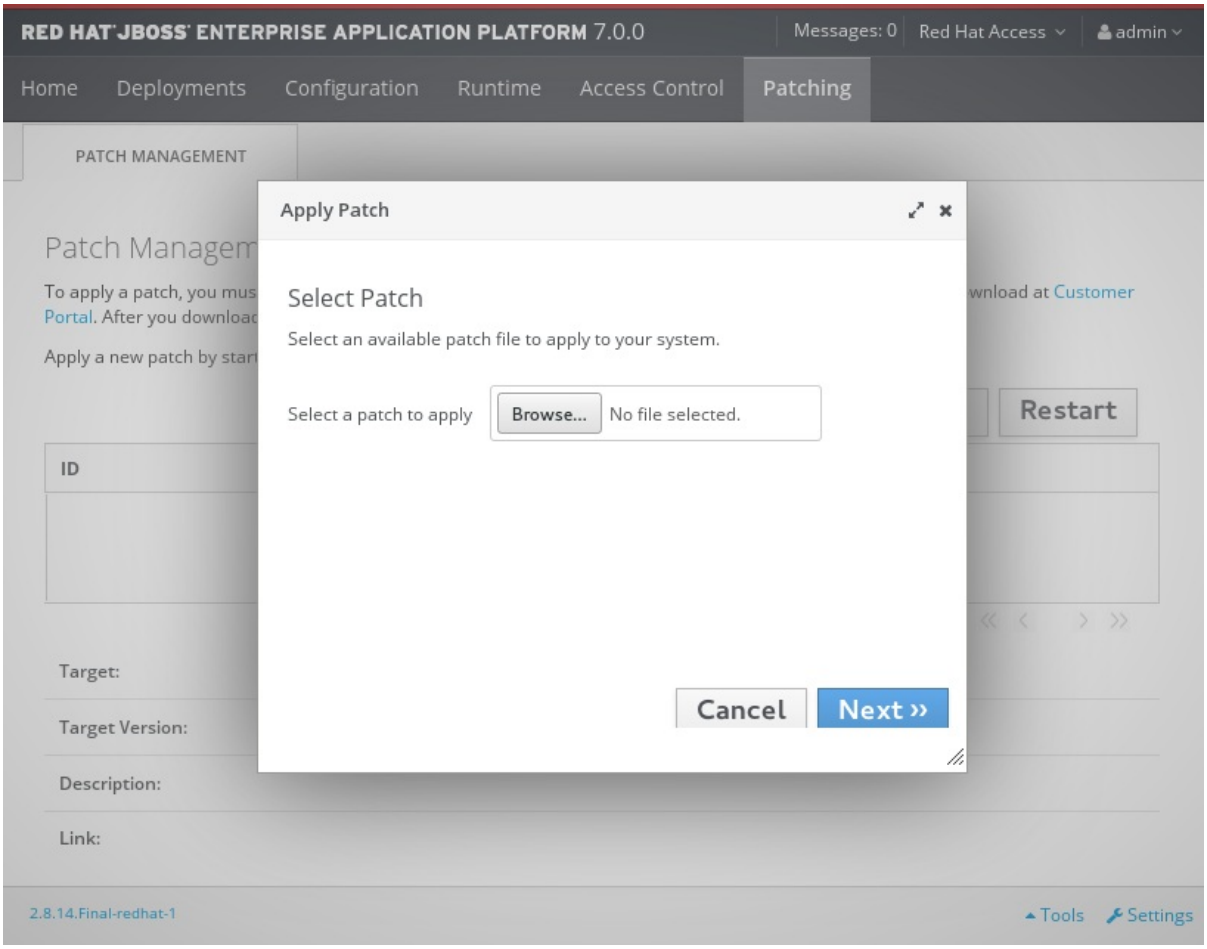
a.

如果您要修补一个受管主机，在下一屏幕中选择是否关闭主机上的服务器，然后单击 **Next**。

4.

点 Browse 按钮，选择您要应用的下载补丁，然后点 Next。

应用补丁屏幕



..如果尝试应用补丁存在冲突，则会显示警告。点 View error details 查看冲突的详情。如果存在冲突，您可以取消操作，或选择 Override all conflicts 复选框，然后单击 Next。覆盖冲突会导致补丁内容覆盖用户的任何修改。

5. 成功应用了补丁后，选择是否立即重启 RH-SSO 以使补丁生效，然后单击" 完成"。

3.2.1.3. 回滚一个补丁

您可以使用 管理 CLI 或管理控制台回滚之前应用的 RH-SSO 补丁。???

重要

使用补丁管理系统回滚补丁并不适合作为常规卸载功能。它仅应在有良好效果的修补程序应用后立即使用。

先决条件

- 之前应用的补丁。

警告

在任一步骤中，在指定 Reset Configuration 选项的值时要小心：

如果设置为 TRUE，则补丁回滚过程也会将 RH-SSO 服务器配置文件回滚到其预补丁状态。应用补丁后对 RH-SSO 服务器配置文件所做的所有更改都将丢失。

如果设置为 FALSE，服务器配置文件将不会回滚。在这种情况下，服务器可能不会在回滚后启动，因为补丁可能会更改了配置，如命名空间，这些配置可能不再有效，必须手动修复。

流程

1. **在管理 CLI 中，使用 patch history 命令查找您要回滚的补丁 ID。**

注意

如果您使用的是受管域，您必须将 --host=*HOSTNAME*参数添加到 此流程中的命令中指定 RH-SSO 主机。

2. **使用上一步中的适当补丁 ID 回滚补丁。**

```
patch rollback --patch-id=PATCH_ID --reset-configuration=TRUE
```

如果尝试回滚补丁时有任何冲突，补丁工具将会发出警告。如果存在冲突，输入 patch --help 来使用可用的参数重新运行命令，并指定如何解决冲突。

3. **重启 RH-SSO 服务器，让补丁回滚来生效：**

```
shutdown --restart=true
```

流程

1.

I.

打开管理控制台，再导航到 Patch Management 视图。

a.

对于单机服务器，请单击 补丁 选项卡。

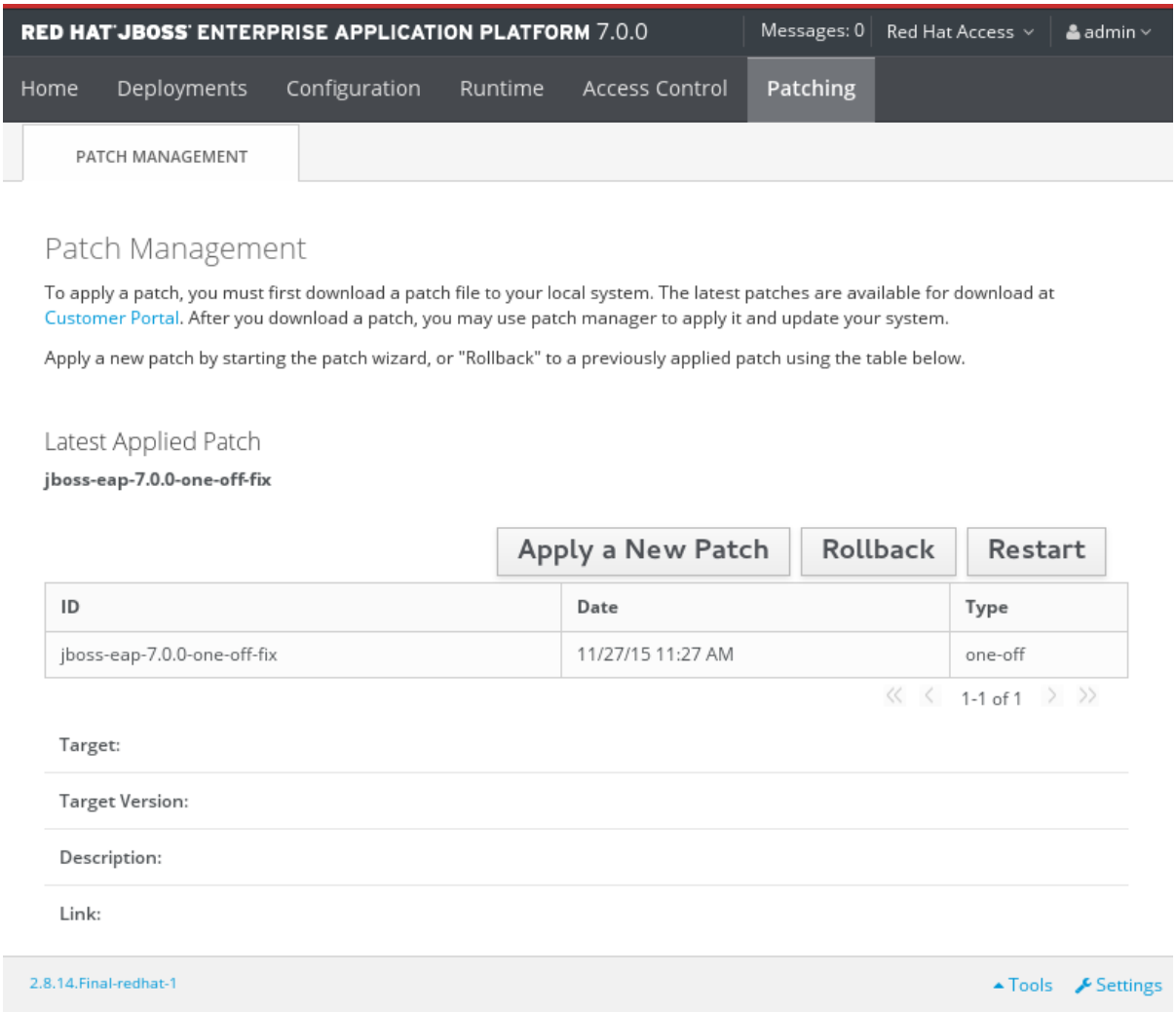
b.

对于受管域中的服务器，单击 patching 选项卡，然后选择要从表中修补的主机，然后单击 View。

2.

从表中列出的标签中选择您要回滚的补丁，然后点 Rollback。

最近的补丁历史记录屏幕

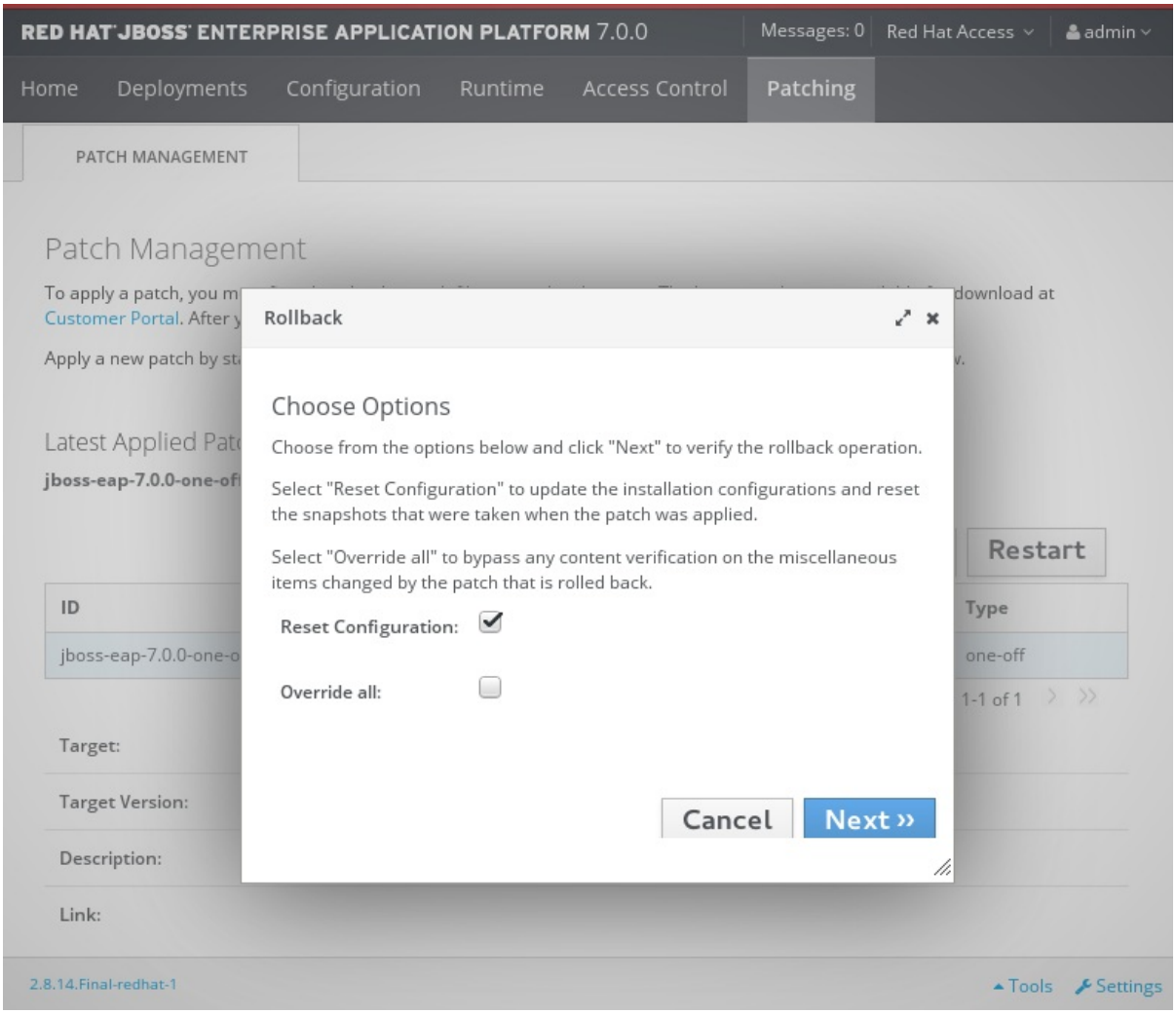


..如果您要在受管集群上回滚补丁，请在下一屏幕中选择是否关闭主机上的服务器，然后单击 Next。

3.

为回滚进程选择您的选项，然后点 Next。

补丁回滚选项



4. 确认选项和要回滚的补丁，然后单击下一步。

   a. 如果尝试回滚补丁并且未选择 覆盖所有 选项存在冲突，则会显示警告。点 View error details 查看冲突的详情。如果存在冲突，您可以取消操作，或者点击 Choose Options，然后使用 覆盖所有 复选框再次尝试操作。覆盖冲突会导致回滚操作覆盖任何用户修改。

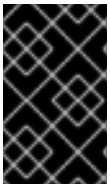
5. 成功回滚补丁后，选择是否现在重启 RH-SSO 服务器以使更改生效，然后单击 Finish。

3.2.1.4. 清除补丁历史记录

当补丁应用到 RH-SSO 服务器时，会保留补丁的内容和补丁历史记录，以便在回滚操作中使用。如果应用了多个累积补丁，补丁历史记录可能会占用大量磁盘空间。

您可以使用以下管理 CLI 命令删除目前尚未使用的所有较早补丁：使用此命令时，只会保留最新的累积补丁和 GA 版本。仅当之前应用了多个累积补丁时，这仅适用于释放空间。

```
/core-service=patching:ageout-history
```



重要

如果您清除了补丁历史记录，您将无法回滚之前应用的补丁。

3.2.2. 修补 RPM 安装

先决条件

- 确保基础操作系统为最新版本，并且已订阅并启用，以便从标准 Red Hat Enterprise Linux 存储库获取更新。
- 确保您已订阅了更新的相关 RH-SSO 存储库。
- 备份所有配置文件、部署和用户数据。

重要

对于受管域，应首先更新 RH-SSO 域控制器。

要从订阅的存储库通过 RPM 安装 RH-SSO 补丁，请使用以下命令更新您的 Red Hat Enterprise Linux 系统：

```
yum update
```

第 4 章 升级 RED HAT SINGLE SIGN-ON 适配器

务必要先升级 Red Hat Single Sign-On 服务器，然后升级适配器。早期版本的适配器可能会用于更新的 Red Hat Single Sign-On 服务器，但早期版本的 Red Hat Single Sign-On 服务器可能无法在以后版本的适配器使用。

4.1. 与旧的适配器兼容

如前文所述，我们尝试支持与较旧版本适配器的 Red Hat Single Sign-On 服务器的新版本。但是，在某些情况下，我们需要在 Red Hat Single Sign-On 服务器端包括修复，这可能会破坏与旧版本适配器的兼容性。例如，在实施 OpenID Connect 规范的新方面时，旧的客户端适配器版本不知道。

在这些情况下，我们添加了兼容性模式。对于 OpenId Connect 客户端，Red Hat Single Sign-On admin 控制台上有一个名为 OpenID Connect Compatibility Modes 的部分，它带有客户端详情。在这里，您可以禁用 Red Hat Single Sign-On 服务器的一些新方面，以保持与旧客户端适配器的兼容性。有关各个交换机的工具提示提供了更多详细信息。

4.2. 升级 EAP 适配器

流程

如果您最初使用下载的归档安装适配器，以升级 JBoss EAP 适配器，请执行以下步骤。

1. 下载新的适配器归档。
2. 通过删除 EAP_HOME/modules/system/add-ons/keycloak/ 目录来删除前面的适配器模块。
3. 将下载的存档解压缩到 EAP_HOME 中。

流程

如果您最初使用 RPM 安装适配器（用于升级适配器），请完成以下步骤，具体根据您是否执行次要还是微型升级而有所不同：

1. 对于次要升级，使用 Yum 卸载您当前安装的任何适配器，然后使用 Yum 安装新版适配器。

2. 对于微升级，请使用 Yum 来升级适配器。这是微升级的唯一步骤。

```
yum update
```

4.3. 升级 JAVASCRIPT 适配器

要升级复制到 web 应用程序的 JavaScript 适配器，请执行以下步骤。

流程

1. 下载新的适配器归档。

2. 使用下载的存档中的 keycloak.js 文件覆盖应用中的 keycloak.js 文件。

4.4. 升级 NODE.JS 适配器

要升级已复制到 web 应用程序的 Node.js 适配器，请执行以下步骤。

流程

1. 下载新的适配器归档。

2. 删除现有 Node.js 适配器目录

3. 将更新的文件解压缩到位

4. 在应用程序的 package.json 中更改 keycloak-connect 的依赖项