# Red Hat Enterprise Linux 8

## 配置和管理身份管理

登录到 IdM 并管理服务、用户、主机、组、访问控制规则和证书。

Last Updated: 2024-07-14

# Red Hat Enterprise Linux 8 配置和管理身份管理

登录到 IdM 并管理服务、用户、主机、组、访问控制规则和证书。

## 法律通告



## 摘要

红帽身份管理(IdM)的主要功能是管理用户、组、主机、访问控制规则和证书。但是，在 IdM 中执行管理任务前，您必须登录到服务。当您使用命令行或 IdM Web UI 登录时，您可以使用 Kerberos 和一次性密码作为 IdM 中的身份验证方法。 您可以使用集成的或外部证书颁发机构(CA)来管理 IdM 中的证书。您可以使用许多工具（如 Ansible Playbook）请求、续订和替换证书。要替换 IdM 服务器的 Web 服务器和 LDAP 服务器证书，您必须执行手动操作。

# 目录

# 使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始： master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。如需了解更多详细信息，请参阅 CTO Chris Wright 信息。

在身份管理中，计划中的术语变化包括：

- 使用 ***block list*** 替换 *blacklist*
- 使用 ***allow list*** 替换 *whitelist*
- 使用 ***secondary*** 替换 *slave*
- *master* 会根据上下文被替换为其他更适当的术语:
  - 使用 ***IdM server*** 替换 *IdM master*
  - 使用 ***CA renewal server*** 替换 *CA renewal master*
  - 使用 ***CRL publisher server*** 替换 *CRL master*
  - 使用 ***multi-supplier*** 替换 *multi-master*

# 对红帽文档提供反馈

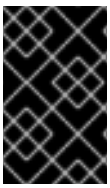
我们感谢您对我们文档的反馈。让我们了解如何改进它。

**通过 Jira 提交反馈（需要帐户）**

1. 登录到 Jira 网站。
2. 单击顶部导航栏中的 **Create**。
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

# 第 1 章 从命令行登录到身份管理

身份管理(IdM)使用 Kerberos 协议来支持单点登录。单点登录意味着用户仅输入一次正确的用户名和密码，就可以访问 IdM 服务，而无需系统再次提示输入凭证。



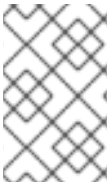
**重要**

在 IdM 中，系统安全服务守护进程(SSSD)在用户成功登录到带有相应 Kerberos 主体名的 IdM 客户端机器上的桌面环境后，会自动为用户获取票据授予票(TGT)。这意味着登录后，用户不需要使用 **kinit** 工具来访问 IdM 资源。

如果您已清除 Kerberos 凭证缓存或者 Kerberos TGT 已过期，您需要手动请求 Kerberos ticket 以访问 IdM 资源。以下章节介绍了在 IdM 中使用 Kerberos 的基本用户操作。

## 1.1. 使用 KINIT 手动登录到 IDM

按照以下流程，使用 **kinit** 工具手动向身份管理(IdM)环境进行身份验证。**kinit** 工具代表 IdM 用户获取并缓存 Kerberos 票据授予票(TGT)。



**注意**

只有在初始 Kerberos TGT 被销毁了或者过期了，才使用这个流程。作为 IdM 用户，当登录到本地机器时，您也会自动登录到 IdM。这意味着登录后，您不需要使用 **kinit** 工具来访问 IdM 资源。

**流程**

1. 要登录到 IdM

   - 在当前登录到本地系统的用户的用户名下，使用 **kinit**，而不指定用户名。例如，如果您在本地系统中以 **example_user** 身份登录：

     ```
     [example_user@server ~]$ kinit
     Password for example_user@EXAMPLE.COM:
     [example_user@server ~]$
     ```

     如果本地用户的用户名与 IdM 中的任何用户条目都不匹配，则身份验证尝试失败：

     ```
     [example_user@server ~]$ kinit
     kinit: Client 'example_user@EXAMPLE.COM' not found in Kerberos database while
     getting initial credentials
     ```

   - 使用与本地用户名不匹配的 Kerberos 主体，将所需的用户名传递给 **kinit** 工具。例如，要以 **admin** 用户身份登录：

     ```
     [example_user@server ~]$ kinit admin
     Password for admin@EXAMPLE.COM:
     [example_user@server ~]$
     ```

2. 另外，要验证登录是否成功，请使用 **klist** 工具来显示缓存的 TGT。在以下示例中，缓存包含了 **example_user** 主体的票，这意味着在这个特定的主机上，当前只允许 **example_user** 访问 IdM 服务：

```
$ klist
Ticket cache: KEYRING:persistent:0:0
Default principal: example_user@EXAMPLE.COM

Valid starting     Expires            Service principal
11/10/2019 08:35:45  11/10/2019 18:35:45  krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

## 1.2. 销毁用户的活动的 KERBEROS 票

按照以下流程清除包含用户的活跃 Kerberos 票据的凭证缓存。

**流程**

1. 销毁您的 Kerberos 票：

```
[example_user@server ~]$ kdestroy
```

2. （可选）检查 Kerberos 票是否已被销毁：

```
[example_user@server ~]$ klist
klist: Credentials cache keyring 'persistent:0:0' not found
```

## 1.3. 为 KERBEROS 身份验证配置外部系统

按照以下流程配置外部系统，以便身份管理(IdM)用户可以使用他们的 Kerberos 凭证从外部系统登录到 IdM。

当您的基础架构包含多个域或重叠域时，在外部系统上启用 Kerberos 身份验证非常有用。如果系统尚未通过 **ipa-client-install** 注册到任何 IdM 域，它也很有用。

要从不属于 IdM 域成员的系统启用对 IdM 的 Kerberos 身份验证，请在外部系统上定义特定于 IdM 的 Kerberos 配置文件。

**先决条件**

- **krb5-workstation** 软件包已安装在外部系统上。
  要查找是否安装了该软件包，请使用以下 CLI 命令：

```
# yum list installed krb5-workstation
Installed Packages
krb5-workstation.x86_64    1.16.1-19.el8     @BaseOS
```

**流程**

1. 将 **/etc/krb5.conf** 文件从 IdM 服务器复制到外部系统。例如：

```
# scp /etc/krb5.conf root@externalsystem.example.com:/etc/krb5_ipa.conf
```

**警告**

不要覆盖外部系统上现有的 **krb5.conf** 文件。

2. 在外部系统上，将终端会话设置为使用复制的 IdM Kerberos 配置文件：

```
$ export KRB5_CONFIG=/etc/krb5_ipa.conf
```

**KRB5_CONFIG** 变量仅在退出之前暂时存在。要防止其丢失，请使用其他文件名导出变量。

3. 将 Kerberos 配置代码段从 **/etc/krb5.conf.d/** 目录复制到外部系统。

外部系统上的用户现在可以使用 **kinit** 工具对 IdM 服务器进行身份验证。

## 1.4. 其它资源

- **krb5.conf(5)** 手册页。
- **kinit(1)** 手册页。
- **klist(1)** 手册页。
- **kdestroy(1)** 手册页。

# 第 2 章 查看、启动和停止身份管理服务

身份管理(IdM)服务器是作为域控制器(DC)的 Red Hat Enterprise Linux 系统。很多不同的服务在 IdM 服务器上运行，最重要的是目录服务器、证书颁发机构(CA)、DNS 和 Kerberos。

## 2.1. IDM 服务

有许多不同服务可以在 IdM 服务器和客户端上安装并运行。

### IdM 服务器托管的服务列表

以下大多数服务并没严格要求安装到 IdM 服务器上。例如，您可以在 IdM 域外的外部服务器上安装诸如证书颁发机构(CA)或 DNS 服务器等服务。

**Kerberos**

**krb5kdc** 和 **kadmin** 服务

IdM 使用 **Kerberos** 协议来支持单点登录。使用 Kerberos，用户只需提供一次正确的用户名和密码，就可以访问 IdM 服务，而系统不需要再次提示输入凭证。

Kerberos 分为两部分：

- **krb5kdc** 服务是 Kerberos 身份验证服务和密钥分发中心(KDC)守护进程。
- **kadmin** 服务是 Kerberos 数据库管理程序。

有关如何在 IdM 中使用 Kerberos 进行身份验证的详情，请参考 从命令行登录到身份管理 和在 Web UI 中登录到 IdM：使用 Kerberos 票据。

**LDAP 目录服务器**

**dirsrv** 服务

IdM **LDAP 目录服务器** 实例存储所有 IdM 信息，例如，与 Kerberos、用户帐户、主机条目、服务、策略、DNS 等相关的信息。LDAP 目录服务器实例基于与 红帽目录服务器 相同的技术。但是，它被调优为特定于 IdM 的任务。

**证书颁发机构**

**pki-tomcatd** 服务

集成的 **证书颁发机构(CA)** 基于与 与红帽证书系统 相同的技术。**pki** 是用于访问证书系统服务的命令行界面。

如果您单独创建并提供了所有必需的证书，则您还可以安装没有集成 CA 的服务器。

如需更多信息，请参阅 规划您的 CA 服务。

**域名系统(DNS)**

**named** 服务

IdM 使用 **DNS** 进行动态服务发现。IdM 客户端安装工具可使用 DNS 的信息来自动配置客户端机器。客户端注册到 IdM 域后，它使用 DNS 来定位域中的 IdM 服务器和服务。Red Hat Enterprise Linux 中的 DNS（域名系统）协议的 **BIND** （Berkeley 互联网名称域）实现包括 **命名的** DNS 服务器。**named-pkcs11** 是使用对 PKCS#11 加密标准的原生支持构建的 BIND DNS 服务器版本。

如需更多信息，请参阅 规划 DNS 服务和主机名。

Apache HTTP 服务器

**httpd** 服务

Apache HTTP Web 服务器提供了 IdM Web UI，还管理证书颁发机构和其他 IdM 服务之间的通信。

Samba/ Winbind

**SMB** 和 **winbind** 服务

Samba 在 Red Hat Enterprise Linux 中实现了服务器消息块(SMB)协议，也称为通用互联网文件系统(CIFS)协议。通过 smb 服务，SMB 协议可让您访问服务器上的资源，如文件共享和共享打印机。如果您使用活动目录(AD)环境配置了信任，'Winbind' 服务将管理 IdM 服务器和 AD 服务器之间的通信。

一次性密码(OTP)验证

**ipa-otpd** 服务

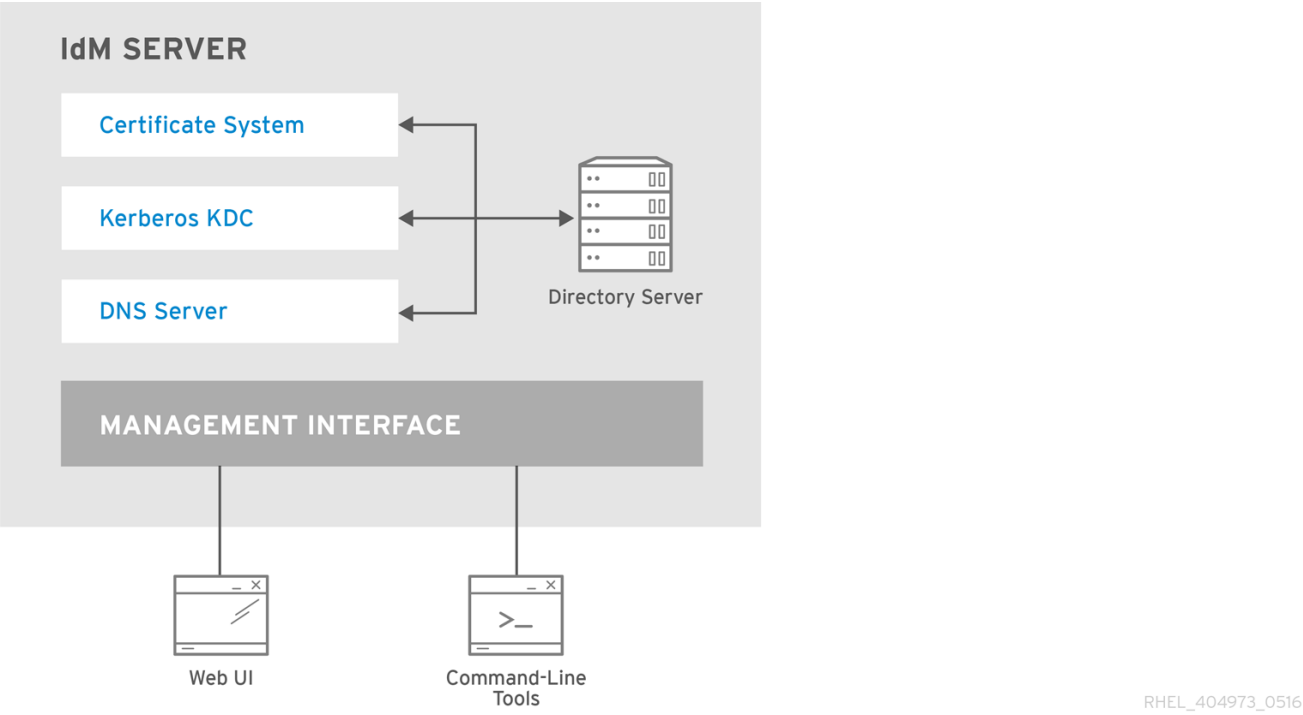
一次性密码(OTP)是由身份验证令牌为一个会话生成的密码，作为双因素身份验证的一部分。OTP 身份验证在 Red Hat Enterprise Linux 中是通过 **ipa-otpd** 服务实现的。

如需更多信息，请参阅 使用一次性密码登录到身份管理 Web UI。

OpenDNSSEC

**ipa-dnskeysyncd** 服务

OpenDNSSEC 是一个 DNS 管理器，自动化了跟踪 DNS 安全扩展(DNSSEC)密钥和区域签名的过程。**ipa-dnskeysyncd** 服务管理 IdM 目录服务器和 OpenDNSSEC 之间的同步。




IdM 客户端托管的服务列表

- **系统安全服务守护进程** ：**sssd** 服务

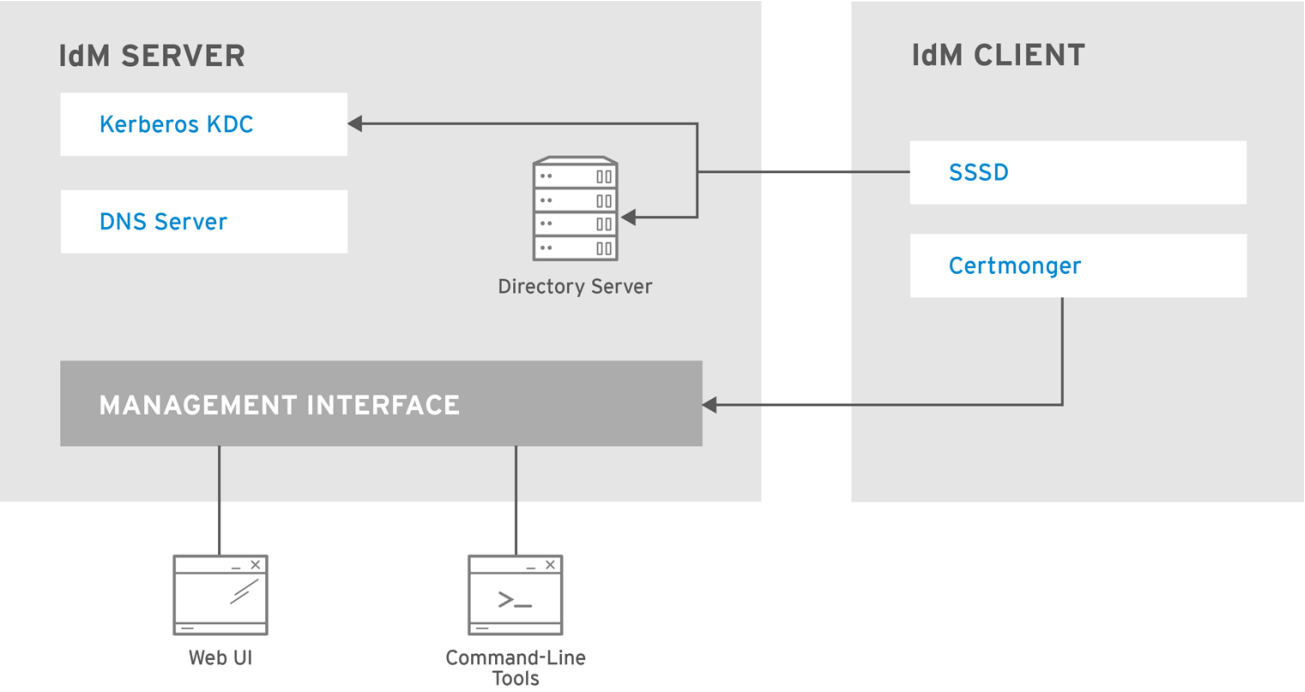
**系统安全服务守护进程** (SSSD)是客户端应用程序，其管理用户身份验证和缓存凭据。缓存可让本地系统在 IdM 服务器不可用或客户端离线时能够继续正常的身份验证操作。

如需更多信息，请参阅了解 SSSD 及其优势。

- Certmonger ：**certmonger** 服务

**certmonger** 服务监控并更新客户端上的证书。它可以为系统上的服务请求新的证书。

如需更多信息，请参阅 使用 certmonger 为服务获取 IdM 证书 。




## 2.2. 查看 IDM 服务的状态

要查看 IdM 服务器上配置的 IdM 服务的状态，请运行 **ipactl status** 命令：

```
[root@server ~]# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
pki-tomcatd Service: RUNNING
smb Service: RUNNING
winbind Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

服务器上 **ipactl status** 命令的结果取决于您的 IdM 配置。例如，如果 IdM 部署不包含 DNS 服务器，则 **named** 服务不会出现在列表中。

**注意**

您不能使用 IdM Web UI 来查看在特定 IdM 服务器上运行的所有 IdM 服务的状态。可以在 IdM Web UI 的 **Identity → Services** 选项卡中查看在不同服务器上运行的 Kerberized 服务。

您可以启动或停止整个服务器，或仅单个服务。

要启动、停止或重启整个 IdM 服务器，请参阅：

- 启动和停止整个身份管理服务器

要启动、停止或重启单个 IdM 服务，请参阅：

- 启动和停止单个身份管理服务

要显示 IdM 软件的版本，请参阅：

- 显示 IdM 软件版本的方法

## 2.3. 启动和停止整个身份管理服务器

使用 **ipa** systemd 服务停止、启动或重启整个 IdM 服务器以及所有安装的服务。使用 **systemctl** 实用程序控制 **ipa** systemd 服务，确保所有服务都以适当的顺序停止、启动或重启。**ipa** systemd 服务也会在启动 IdM 服务前升级 RHEL IdM 配置，并在管理 IdM 服务时使用正确的 SELinux 上下文。您不需要具有有效的 Kerberos 票据来运行 **systemctl ipa** 命令。

### ipa systemd service 命令

启动整个 IdM 服务器：

```
# systemctl start ipa
```

停止整个 IdM 服务器：

```
# systemctl stop ipa
```

重启整个 IdM 服务器：

```
# systemctl restart ipa
```

要显示组成 IdM 的所有服务的状态，请使用 **ipactl** 工具：

```
# ipactl status
```

**重要**

- 不要直接使用 **ipactl** 工具来启动、停止或重启 IdM 服务。使用 **systemctl ipa** 命令，在可预测的环境中调用 **ipactl** 工具。
- 您不能使用 IdM Web UI 来执行 **ipactl** 命令。

## 2.4. 启动和停止单个身份管理服务

通常不建议手动更改 IdM 配置文件。然而，在某些情况下，需要管理员来执行特定服务的手动配置。在这种情况下，使用 **systemctl** 工具来停止、启动或重启单个 IdM 服务。

例如，自定义目录服务器行为，而不修改其他 IdM 服务后使用 **systemctl**：

```
# systemctl restart dirsrv@REALM-NAME.service
```

另外，在最初使用活动目录部署 IdM 信任时，请修改 **/etc/sssd/sssd.conf** 文件，并添加：

- 在远程服务器具有高延迟的环境中调整超时配置选项的特定参数
- 用于调整活动目录站点关联性的特定参数
- 覆盖不是由全局 IdM 设置提供的某些配置选项

要应用您在 **/etc/sssd/sssd.conf** 文件中所做的更改：

```
# systemctl restart sssd.service
```

需要运行 **systemctl restart sssd.service**，因为系统安全服务守护进程(SSSD)不会自动重新读取或重新应用其配置。

请注意，对于影响 IdM 身份范围的更改，建议完全重启服务器。

**重要**

要重启多个 IdM 域服务，请始终使用 **systemctl restart ipa**。由于与 IdM 服务器一起安装的服务之间的依赖关系，这些服务启动和停止的顺序至关重要。**ipa** systemd 服务确保服务以适当的顺序启动和停止。

### 有用的 systemctl 命令

要启动特定的 IdM 服务：

```
# systemctl start name.service
```

要停止特定的 IdM 服务：

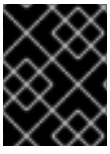
```
# systemctl stop name.service
```

要重启特定的 IdM 服务：

```
# systemctl restart name.service
```

要查看特定的 IdM 服务的状态：

```
# systemctl status name.service
```



**重要**

您不能使用 IdM Web UI 来启动或停止在 IdM 服务器上运行的单个服务。您只能使用 Web UI 来修改 Kerberized 服务的设置，方法是导航到 **Identity → Services**，并选择服务。

### 其它资源

- 启动和停止整个身份管理服务器

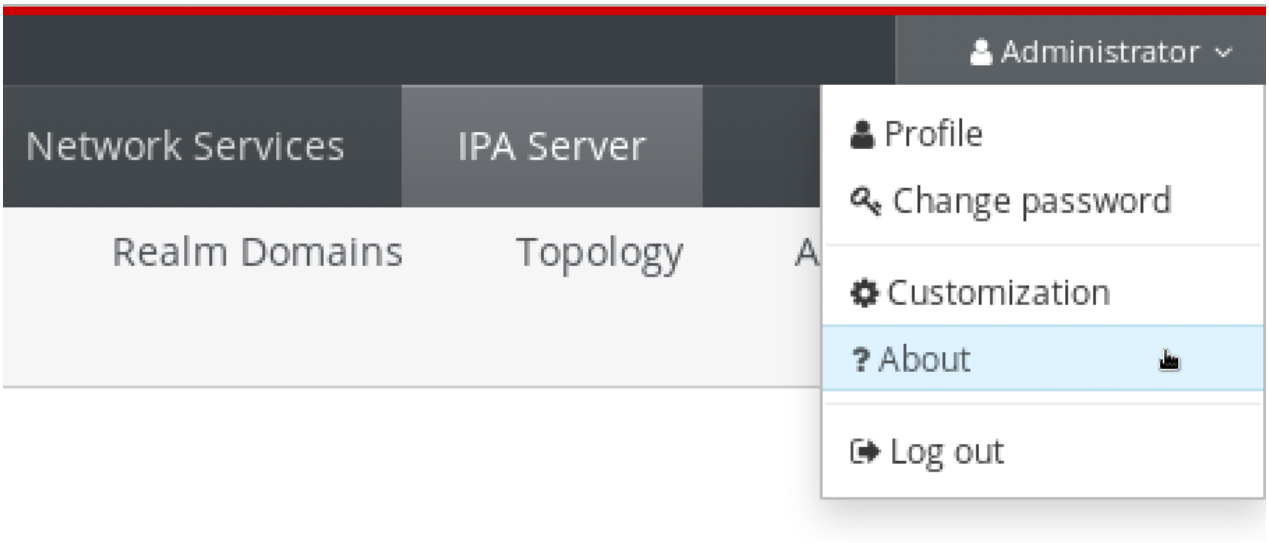
## 2.5. 显示 IDM 软件版本的方法

您可以使用以下命令显示 IdM 版本号：

- The IdM WebUI
- **ipa** 命令
- **rpm** 命令

**通过 WebUI 显示版本**

在 IdM Web UI 中，可以通过从右上角的用户名菜单中选择 **About** 来显示软件版本。




**使用 ipa 命令显示版本**

在命令行中使用 **ipa --version** 命令。

```
[root@server ~]# ipa --version
VERSION: 4.8.0, API_VERSION: 2.233
```

**使用 rpm 命令显示版本**

如果 IdM 服务工作不正常，您可以使用 **rpm** 工具来确定当前安装的 **ipa-server** 软件包的版本号。

```
[root@server ~]# rpm -q ipa-server
ipa-server-4.8.0-11.module+el8.1.0+4247+9f3fd721.x86_64
```

# 第 3 章 IDM 命令行工具简介

了解有关使用身份管理(IdM)命令行工具的基础知识。

**先决条件**

- 已安装并可访问 IdM 服务器。
  详情请参阅 安装身份管理。
- 要使用 IPA 命令行界面，请使用有效的 Kerberos 票向 IdM 进行身份验证。
  有关获取有效的 Kerberos 票据的详情，请参阅 从命令行登录到身份管理。

## 3.1. 什么是 IPA 命令行界面

IPA 命令行界面(CLI)是身份管理(IdM)管理的基本命令行界面。

它支持很多管理 IdM 的子命令，如 **ipa user-add** 命令来添加新用户。

IPA CLI 允许您：

- 在网络中添加、管理或删除用户、组、主机和其他对象。
- 管理证书。
- 搜索条目。
- 显示和列出对象。
- 设置访问权限。
- 获取正确命令语法的帮助。

## 3.2. IPA 帮助是什么

IPA 帮助是 IdM 服务器的内置文档系统。

IPA 命令行界面(CLI)从加载的 IdM 插件模块中生成可用的帮助主题。要使用 IPA 帮助工具，您必须：

- IdM 服务器已安装并运行。
- 使用有效的 Kerberos 票据进行了身份验证。

输入没有选项的 **ipa help** 命令会显示有关基本帮助用法和最常见的命令示例的信息。

您可以对不同的 **ipa help** 用例使用以下选项：

```
$ ipa help [TOPIC | COMMAND | topics | commands]
```

- **[]** - 方括号表示所有参数都是可选的，您可以只写 **ipa help**，命令就可执行。
- **|** - 管道符表示 **或** 。因此，您可以使用基本的 **ipa help** 命令指定 **TOPIC**、**COMMAND** 或 **topics**、**commands** ：
  - **topics** – 您可以运行命令 **ipa help topics** 来显示 IPA 帮助涵盖的主题列表，如 **user**、**cert**、**server** 等。

- **TOPIC** – 大写字母的 TOPIC 是一个变量。因此，您可以指定一个特定的主题，例如 **ipa help user**。
- **commands** – 您可以输入命令 **ipa help commands** 来显示 IPA 帮助所涵盖的命令列表，如 **user-add**、**ca-enable**、**server-show** 等。
- **COMMAND** – 大写字母的 COMMAND 是一个变量。因此，您可以指定一个的命令，例如 **ipa help user-add**。

## 3.3. 使用 IPA 帮助主题

以下流程描述了如何在命令行界面中使用 IPA 帮助。

**流程**

1. 打开一个终端，接到 IdM 服务器。
2. 输入 **ipa help topics** 来显示帮助所涵盖的主题列表。

```
$ ipa help topics
```

3. 选择其中一个主题，按照以下模式创建一个命令： **ipa help *[topic_name]***。添加在上一步中列出的主题之一，而不是 **topic_name** 字符串。
在这个示例中，我们使用以下主题：**user**

```
$ ipa help user
```

4. 如果 IPA 帮助输出太长，您不能整个文本，请用以下语法：

```
$ ipa help user | less
```

然后您可以向下滚动，并阅读全部帮助。

IPA CLI 显示 **user** 主题的帮助页。阅读完概述后，您可以看到许多使用主题命令的模式示例。

## 3.4. 使用 IPA HELP 命令

以下流程描述了如何在命令行界面中创建 IPA 帮助命令。

**流程**

1. 打开一个终端，接到 IdM 服务器。
2. 输入 **ipa help commands** 来显示 help 所涵盖的命令列表。

```
$ ipa help commands
```

3. 选择一个命令，并按照下模式创建一个help 命令： **ipa help *<COMMAND>***。添加在上一步中列出的其中一个命令，而不是 **<COMMAND>** 字符串。

```
$ ipa help user-add
```

**其它资源**

- **ipa** 手册页。

## 3.5. IPA 命令的结构

IPA CLI 区分以下命令类型：

- **内置命令** – IdM 服务器中可用的内置命令。
- **插件提供的命令**

IPA 命令的结构允许您管理各种类型的对象。例如：

- 用户，
- 主机、
- DNS 记录、
- 证书、

以及许多其他信息。

对于大多数这些对象，IPA CLI 包括以下命令来：

- 添加（**add**）
- 修改(**mod**)
- 删除(**del**)
- 搜索（**find**）
- 显示（**show**）

命令具有以下结构：

**ipa user-add**, **ipa user-mod**, **ipa user-del**, **ipa user-find**, **ipa user-show**

**ipa host-add**, **ipa host-mod**, **ipa host-del**, **ipa host-find**, **ipa host-show**

**ipa dnsrecord-add**, **ipa dnsrecord-mod**, **ipa dnsrecord-del**, **ipa dnsrecord-find**, **ipa dnrecord-show**

您可以使用 **ipa user-add [options]** 创建用户，其中 **[options]** 是可选的。如果您只使用 **ipa user-add** 命令，脚本将逐个询问您详细信息。

要更改现有对象，您需要定义对象，因此命令还包括一个对象：**ipa user-mod USER_NAME [options]**。

## 3.6. 使用 IPA 命令将用户帐户添加到 IDM

以下流程描述了如何使用命令行向身份管理(IdM)数据库添加一个新用户。

**先决条件**

- 您需要拥有管理员特权才能将用户帐户添加到 IdM 服务器。

**流程**

1. 打开一个终端，接到 IdM 服务器。
2. 输入命令来添加新用户：

```
$ ipa user-add
```

命令运行一个脚本，提示您提供创建用户帐户所需的基本数据。

3. 在 **First name:** 字段中，输入新用户的名字，然后按 **Enter** 键。
4. 在 **Last name:** 字段中，输入新用户的姓氏，然后按 **Enter** 键。
5. 在 **User login [suggested user name]:**输入用户名，或者只是按 **Enter** 键来接受推荐的用户名。
整个 IdM 数据库的用户名必须是唯一的。如果因为用户名已存在而发生错误，使用 **ipa user-add** 命令重复该过程，并使用不同的、唯一的用户名。

添加用户名后，用户帐户被添加到 IdM 数据库，IPA 命令行界面(CLI)会打印以下输出：

```
----------------------
Added user "euser"
----------------------
User login: euser
First name: Example
Last name: User
Full name: Example User
Display name: Example User
Initials: EU
Home directory: /home/euser
GECOS: Example User
Login shell: /bin/sh
Principal name: euser@IDM.EXAMPLE.COM
Principal alias: euser@IDM.EXAMPLE.COM
Email address: euser@idm.example.com
UID: 427200006
GID: 427200006
Password: False
Member of groups: ipausers
Kerberos keys available: False
```

**注意**

默认情况下，没有为用户帐户设置用户密码。要在创建用户帐户时添加密码，请使用带有以下语法的 **ipa user-add** 命令：

```
$ ipa user-add --first=Example --last=User --password
```

然后 IPA CLI 会提示您添加或确认用户名和密码。

如果已创建了该用户，您可以使用 **ipa user-mod** 命令添加密码。

**其它资源**

- 运行 **ipa help user-add** 命令来了解有关参数的更多信息。

## 3.7. 使用 IPA 命令修改 IDM 中的用户帐户

您可以为每个用户帐户更改多个参数。例如，您可以为用户添加新密码。

基本命令语法与 **user-add** 语法不同，因为您需要定义要对其执行更改的现有用户帐户，例如，添加密码。

**先决条件**

- 您需要具有管理员特权才能修改用户帐户。

**流程**

1. 打开一个终端，接到 IdM 服务器。
2. 输入 **ipa user-mod** 命令，指定要修改的用户，以及任何选项，如添加密码的 **--password** ：

   ```
   $ ipa user-mod euser --password
   ```

   命令将运行脚本，您可以在其中添加新密码。
3. 输入新密码并按 **Enter** 键。

IPA CLI 打印以下输出：

```
----------------------
Modified user "euser"
----------------------
User login: euser
First name: Example
Last name: User
Home directory: /home/euser
Principal name: euser@IDM.EXAMPLE.COM
Principal alias: euser@IDM.EXAMPLE.COM
Email address: euser@idm.example.com
UID: 427200006
GID: 427200006
Password: True
Member of groups: ipausers
Kerberos keys available: True
```

现在，为帐户设置了用户密码，用户可以登录 IdM 了。

**其它资源**

- 运行 **ipa help user-mod** 命令来了解有关参数的更多信息。

## 3.8. 如何为 IDM 工具提供值列表

身份管理(IdM)将多值属性的值存储在列表中。

IdM 支持以下提供多值列表的方法：

- 在同一命令调用中多次使用相同的命令行参数：

```
$ ipa permission-add --right=read --permissions=write --permissions=delete ...
```

- 或者，您可以将列表用大括号括起来，在这种情况下，shell 执行展开：

```
$ ipa permission-add --right={read,write,delete} ...
```

上面的示例显示了命令 **permission-add**，它为对象添加权限。示例中没有提及对象。需要添加要为其添加权限的对象，而不是 **...**。

当您从命令行更新此类多值属性时，IdM 会使用新列表完全覆盖以前的值列表。因此，当更新多值属性时，您必须指定整个新列表，而不只是您要添加的单个值。

例如，在以上命令中，权限列表包括读、写和删除。当您决定使用 **permission-mod** 命令更新列表时，您必须添加所有的值，否则未提及的值将被删除。

**示例 1: -ipa permission-mod** 命令更新所有以前添加的权限。

```
$ ipa permission-mod --right=read --right=write --right=delete ...
```

或者

```
$ ipa permission-mod --right={read,write,delete} ...
```

**示例 2 - ipa permission-mod** 命令会删除 **--right=delete** 参数，因为它没有包含在命令中：

```
$ ipa permission-mod --right=read --right=write ...
```

或者

```
$ ipa permission-mod --right={read,write} ...
```

## 3.9. 如何在 IDM 工具中使用特殊字符

将包含特殊字符的命令行参数传递给 **ipa** 命令时，请使用反斜杠(\)转义这些字符。例如，常见的特殊字符包括尖括号（< 和 >）、and(&)、星号(*)或竖线(|)。

例如，要转义星号(*)：

```
$ ipa certprofile-show certificate_profile --out=exported\*profile.cfg
```

包含未转义特殊字符的命令无法按预期工作，因为 shell 无法正确解析这些字符。

# 第 4 章 从命令行搜索身份管理条目

以下章节描述了如何使用 IPA 命令，其可帮助您查找或显示对象。

## 4.1. 列出 IDM 条目的概述

您可以使用 **ipa *-find** 命令帮助您搜索特定类型的 IdM 条目。

要列出所有 **find** 命令，请使用以下 ipa help 命令：

```
$ ipa help commands | grep find
```

您可能需要检查特定的用户是否包含在 IdM 数据库中。然后您可以使用以下命令列出所有用户：

```
$ ipa user-find
```

要列出其指定属性包含关键字的用户组：

```
$ ipa group-find keyword
```

例如，**ipa group-find *admin*** 命令列出了其名称或描述包含字符串 **admin** 的所有组：

```
----------------
3 groups matched
----------------
  Group name: admins
  Description: Account administrators group
  GID: 427200002

  Group name: editors
  Description: Limited admins who can edit other users
  GID: 427200002

  Group name: trust admins
  Description: Trusts administrators group
```

在搜索用户组时，您还可以将搜索结果限制为包含特定用户的组：

```
$ ipa group-find --user=user_name
```

搜索不包含特定用户的组：

```
$ ipa group-find --no-user=user_name
```

## 4.2. 显示特定条目的详情

使用 **ipa *-show** 命令显示特定 IdM 条目的详情。

**流程**

- 要显示名为 *server.example.com* 的主机的详情：

```
$ ipa host-show server.example.com

Host name: server.example.com
Principal name: host/server.example.com@EXAMPLE.COM
...
```

## 4.3. 调整搜索大小和时间限制

有些查询（比如请求 IdM 用户列表）可能会返回大量条目。通过调优这些搜索操作，您可以在运行 **ipa *-find** 命令时提高服务器的总体性能，例如 **ipa user-find**，并在 Web UI 中显示相应的列表。

**搜索大小限制**

定义从客户端 CLI 发送发送到服务器的请求或从访问 IdM Web UI 的浏览器返回的最大条目数。
默认： 100 条目。

**搜索时间限制**

定义服务器等待搜索运行的最长时间（以秒为单位）。搜索达到这个限制后，服务器将停止搜索并返回该时间里发现的条目。
默认： 2 秒。

如果您将值设为 **-1**，IdM 在搜索时不会应用任何限制。

**重要**

如果设置的搜索大小或时间限制太大，则可能会对服务器性能造成负面影响。

### 4.3.1. 在命令行中调整搜索大小和时间限制

以下流程描述了在命令行中调整搜索大小和时间限制：

- 全局
- 对于一个特定条目

**流程**

1. 要在 CLI 中显示当前搜索时间和大小限制，请使用 **ipa config-show** 命令：

```
$ ipa config-show

Search time limit: 2
Search size limit: 100
```

2. 要为所有查询调整 **全局** 限制，请使用 **ipa config-mod** 命令，并添加 **--searchrecordslimit** 和 **--searchtimelimit** 选项。例如：

```
$ ipa config-mod --searchrecordslimit=500 --searchtimelimit=5
```

3. 要仅为特定查询 **暂时** 调整限制，请在命令中添加 **--sizelimit** 或 **--timelimit** 选项。例如：

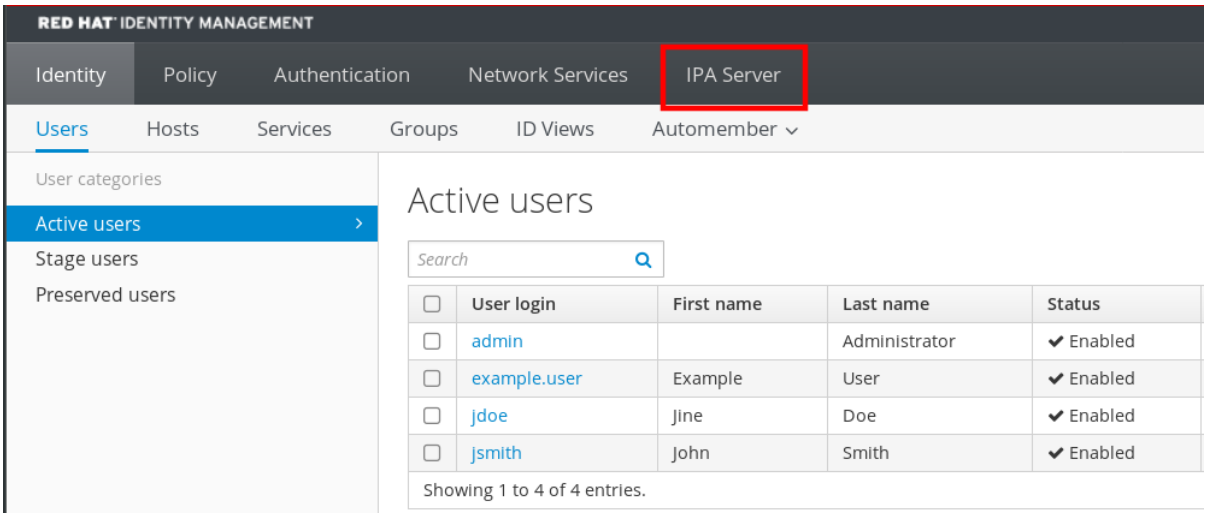
```
$ ipa user-find --sizelimit=200 --timelimit=120
```

### 4.3.2. 在 Web UI 中调整搜索大小和时间限制

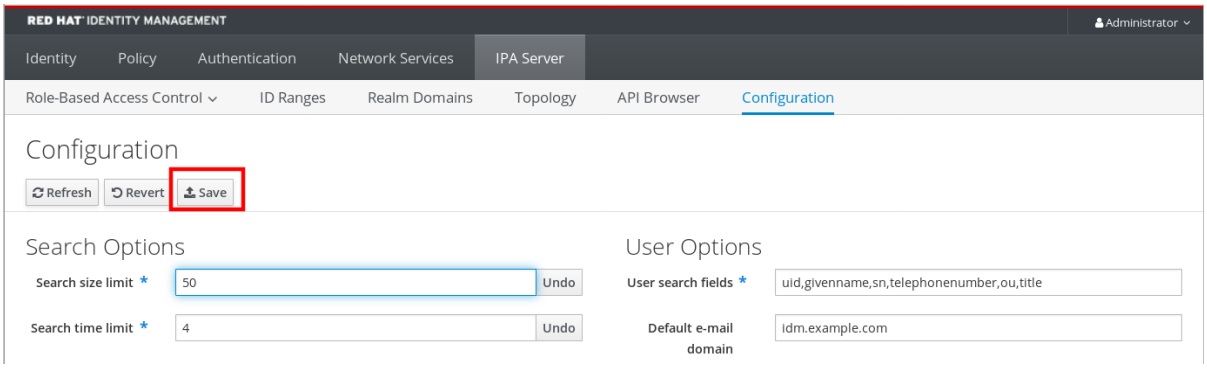
以下流程描述了在 IdM Web UI 中调整全局搜索大小和时间限制。

**流程**

1. 登录到 IdM Web UI。
2. 点 **IPA Server**。




3. 在 **IPA Server** 选项卡中点 **Configuration**。
4. 在**搜索选项**区域中设置所需的值。
   默认值为：
   - 搜索大小限制： 100 个条目
   - 搜索时间限值： 2 秒
5. 点页面顶部的 **Save**。



# 第 5 章 在 WEB 浏览器中访问 IDM WEB UI

IdM （身份管理）Web UI 是一个 IdM 管理的 Web 应用程序，是 IdM 命令行界面(CLI)的图形替代方案。

## 5.1. 什么是 IDM WEB UI

IdM （身份管理）Web UI 是一个 IdM 管理的 Web 应用程序。您可以以以下方式访问 IdM Web UI：

- **IdM 用户：** 有限的一组操作，具体取决于为 IdM 服务器中的用户授予的权限。基本上，活动的 IdM 用户可以登录 IdM 服务器，并配置他们自己的帐户。它们无法更改其他用户的设置或 IdM 服务器的设置。
- **管理员：** 对 IdM 服务器具有完整访问权限。
- **活动用户：** 一组操作，具体取决于授予用户的权限。活动目录用户现在可以是身份管理的管理员。详情请参阅 启用 AD 用户来管理 IdM。

## 5.2. 支持访问 WEB UI 的 WEB 浏览器

身份管理(IdM)支持以下浏览器来连接到 Web UI：

- Mozilla Firefox 38 及更新的版本
- Google Chrome 46 及更新的版本

**注意**

如果您的浏览器尝试使用 TLS v1.3，您可能会遇到使用智能卡访问 IdM Web UI 的问题。

```
[ssl:error] [pid 125757:tid 140436077168384] [client 999.999.999.999:99999] AH:
verify client post handshake
[ssl:error] [pid 125757:tid 140436077168384] [client 999.999.999.999:99999]
AH10158: cannot perform post-handshake authentication
[ssl:error] [pid 125757:tid 140436077168384] SSL Library Error: error:14268117:SSL
routines:SSL_verify_client_post_handshake:extension not received
```

这是因为，最新的浏览器版本没有默认启用 TLS Post-Handshake Authentication(PHA)，或者不支持 PHA。对于网站的一部分，PHA 只需要 TLS 客户端证书，比如使用智能卡验证访问 IdM Web UI 时。

要在 Mozilla Firefox 68 及更新的版本中解决这个问题，请启用 TLS PHA：

1. 在地址栏中输入 **about:config** 以访问 Mozilla Firefox 首选项菜单。
2. 在搜索栏中输入 **security.tls.enable_post_handshake_auth**。
3. 点切换按钮将参数设置为 true。

要解决 Chrome （目前不支持 PHA）的问题，请禁用 TLS v1.3：

1. 打开 **/etc/httpd/conf.d/ssl.conf** 配置文件。
2. 将 **-TLSv1.3** 添加到 **SSLProtocol** 选项：

```
SSLProtocol all -TLSv1 -TLSv1.1 -TLSv1.3
```

3. 重启 **httpd** 服务：

```
service httpd restart
```

请注意，IdM 管理 **ssl.conf** 文件，并可能会在软件包更新过程中覆盖其内容。在更新 IdM 软件包后验证自定义设置。

## 5.3. 访问 WEB UI

以下流程描述了首次使用密码登录到 IdM（身份管理）Web UI。

第一次登录后，您可以将 IdM 服务器配置为使用以下方式进行身份验证：

- Kerberos 票据
  详情请查看 身份管理中的 Kerberos 验证。
- 智能卡
  详情请参阅 为智能卡身份验证配置 IdM 服务器。
- 一次性密码(OTP) - 可将其与密码和 Kerberos 身份验证结合使用。
  详情请参阅 身份管理中的一次性密码(OTP)身份验证。

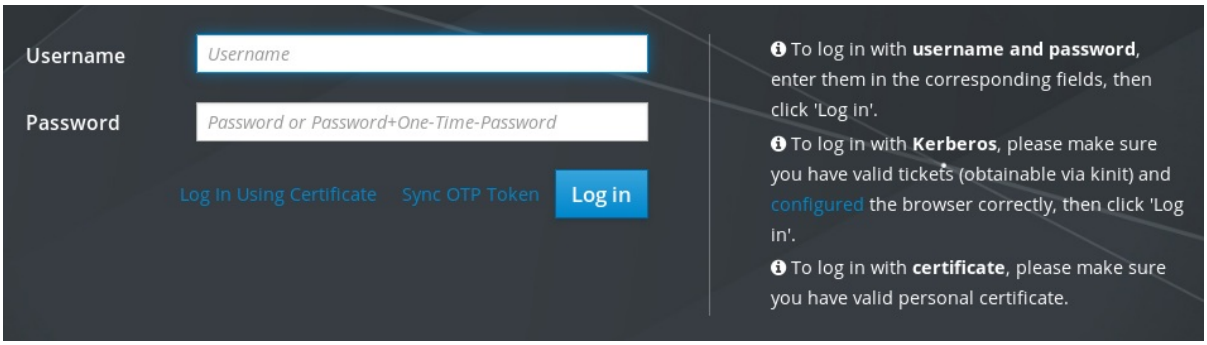
**流程**

1. 在浏览器地址栏中输入 IdM 服务器 URL。名称类似以下示例：

```
https://server.example.com
```

   您只需要将 **server.example.com** 更改为您 IdM 服务器的 DNS 名称。

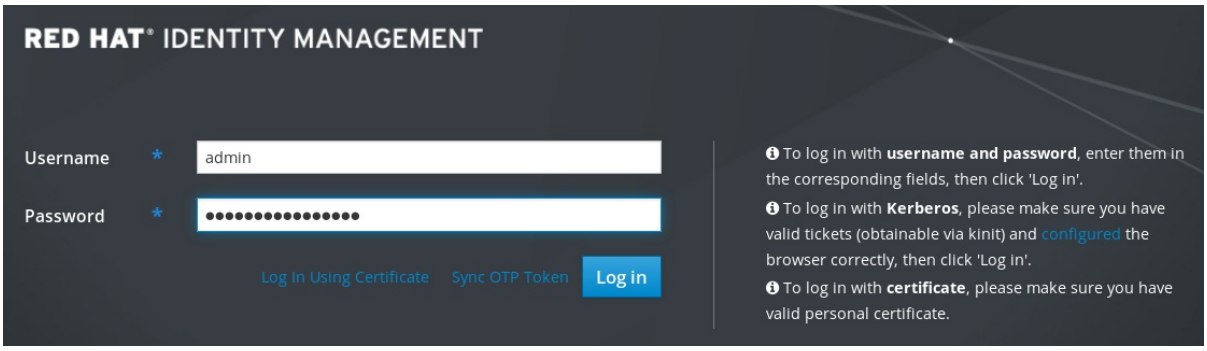
   这会在您的浏览器中打开 IdM Web UI 登录屏幕。




   - 如果服务器没有响应或者登录屏幕没有打开，请检查您要连接的 IdM 服务器上的 DNS 设置。
   - 如果您使用自签名证书，浏览器会发出警告。检查证书并接受安全例外以进行登录。为避免安全异常，请安装由证书颁发机构签名的证书。

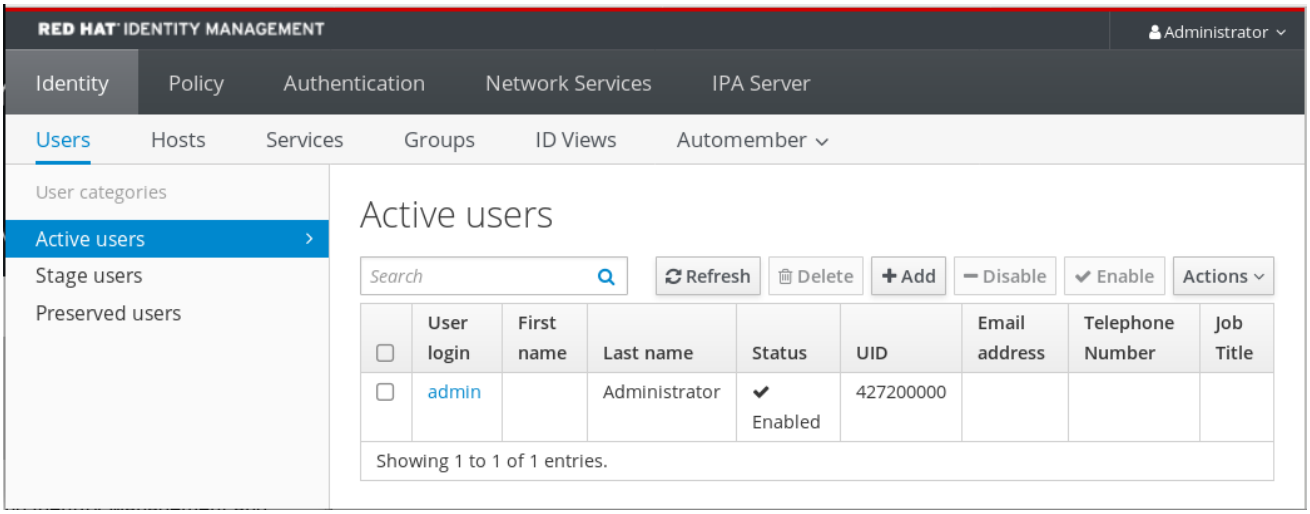
2. 在 Web UI 登录屏幕上，输入您在 IdM 服务器安装过程中添加的管理员帐户凭证。详情请参阅 安装身份管理服务器：使用集成的 DNS，以及集成的 CA。

   如果您已经进入到 IdM 服务器中，您还可以输入您的个人帐户凭证。




3. 单击 **Log in**。

登录成功后，您可以开始配置 IdM 服务器。


RED HAT IDENTITY MANAGEMENT
Administrator
Identity
Policy
Authentication
Network Services
IPA Server
Users
Hosts
Services
Groups
ID Views
Automember
User categories
Active users
Stage users
Preserved users
Active users
Search
Refresh
Delete
Add
Disable
Enable
Actions
User login
First name
Last name
Status
UID
Email address
Telephone Number
Job Title
admin
Administrator
Enabled
427200000
Showing 1 to 1 of 1 entries.

# 第 6 章 在 WEB UI 中登录到 IDM: 使用 KERBEROS 票据

了解更多有关如何配置您的环境，以使用 Kerberos 身份验证，使 Kerberos 能够登录到 IdM Web UI，并访问 IdM。

**先决条件**

- 在网络环境中已安装 IdM 服务器
  详情请参阅 在 Red Hat Enterprise Linux 8 中安装身份管理

## 6.1. 身份管理中的 KERBEROS 身份验证

身份管理(IdM)使用 Kerberos 协议来支持单点登录。单点登录身份验证允许您仅提供一次正确的用户名和密码，然后您就可以访问身份管理服务了，而系统不再提示输入凭据。

如果正确配置了 DNS 和证书设置，IdM 服务器会在安装后立即提供 Kerberos 身份验证。详情请参阅 安装身份管理。

要在主机上使用 Kerberos 身份验证，请安装：

- IdM 客户端
  详情请参阅 为身份管理客户端安装准备系统。
- krb5conf 软件包

## 6.2. 使用 KINIT 手动登录到 IDM

按照以下流程，使用 **kinit** 工具手动向身份管理(IdM)环境进行身份验证。**kinit** 工具代表 IdM 用户获取并缓存 Kerberos 票据授予票(TGT)。

**注意**

只有在初始 Kerberos TGT 被销毁了或者过期了，才使用这个流程。作为 IdM 用户，当登录到本地机器时，您也会自动登录到 IdM。这意味着登录后，您不需要使用 **kinit** 工具来访问 IdM 资源。

**流程**

1. 要登录到 IdM
   - 在当前登录到本地系统的用户的用户名下，使用 **kinit**，而不指定用户名。例如，如果您在本地系统中以 **example_user** 身份登录：

     ```
     [example_user@server ~]$ kinit
     Password for example_user@EXAMPLE.COM:
     [example_user@server ~]$
     ```

     如果本地用户的用户名与 IdM 中的任何用户条目都不匹配，则身份验证尝试失败：

     ```
     [example_user@server ~]$ kinit
     kinit: Client 'example_user@EXAMPLE.COM' not found in Kerberos database while
     getting initial credentials
     ```

- 使用与本地用户名不匹配的 Kerberos 主体，将所需的用户名传递给 **kinit** 工具。例如，要以 **admin** 用户身份登录：

```
[example_user@server ~]$ kinit admin
Password for admin@EXAMPLE.COM:
[example_user@server ~]$
```

2. 另外，要验证登录是否成功，请使用 **klist** 工具来显示缓存的 TGT。在以下示例中，缓存包含了 **example_user** 主体的票，这意味着在这个特定的主机上，当前只允许 **example_user** 访问 IdM 服务：

```
$ klist
Ticket cache: KEYRING:persistent:0:0
Default principal: example_user@EXAMPLE.COM

Valid starting     Expires            Service principal
11/10/2019 08:35:45  11/10/2019 18:35:45  krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

## 6.3. 为 KERBEROS 身份验证配置浏览器

要启用使用 Kerberos 票据的身份验证，您可能需要浏览器配置。

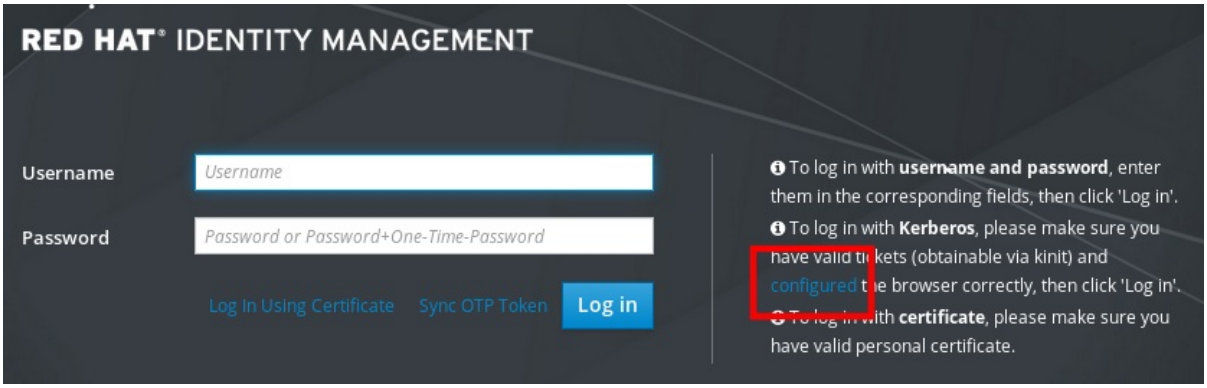
以下步骤可帮助您支持 Kerberos 协商以访问 IdM 域。

每个浏览器支持 Kerberos 的方式不同，并且需要不同的设置。IdM Web UI 包含对以下浏览器的指南：

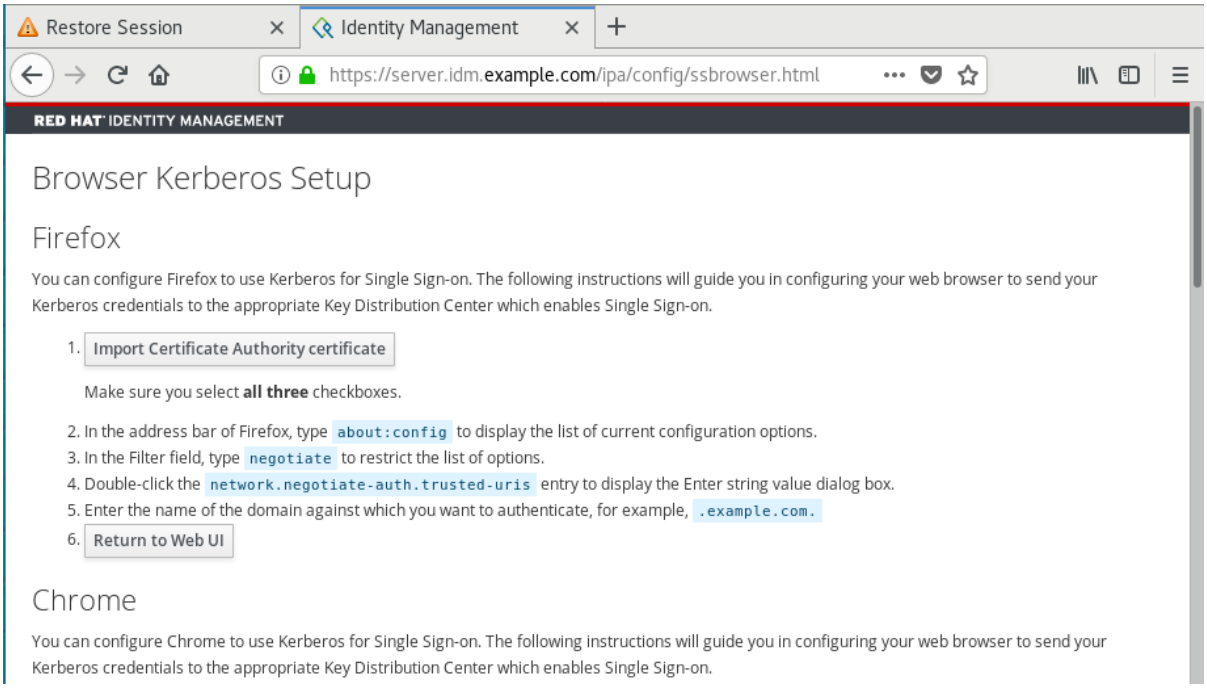
- Firefox
- Chrome

**流程**

1. 在 Web 浏览器中打开 IdM Web UI 登录对话框。
2. 在 Web UI 登录屏幕上，单击浏览器配置的链接。




3. 按照配置页面中的步骤进行操作。




设置完成后，切回到 IdM Web UI，并单击 **Log in**。

## 6.4. 使用 KERBEROS 票据登录到 WEB UI

按照以下流程，使用 Kerberos 票据授予票(TGT)登录到 IdM Web UI。

TGT 在预定义的时间过期。默认的时间间隔为 24 小时，您可以在 IdM Web UI 中更改它。

时间间隔过期后，您需要续订票据：

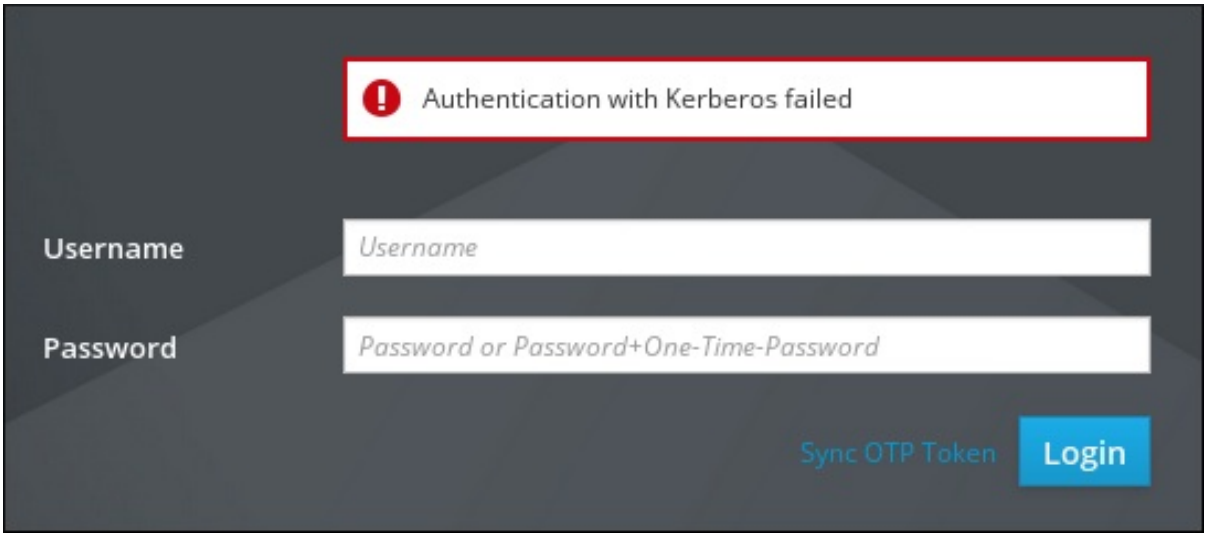
- 使用 kinit 命令。
- 在 Web UI 登录对话框中使用 IdM 登录凭据。

**流程**

- 打开 IdM Web UI。
  如果 Kerberos 身份验证正常工作，并且您拥有有效的票据，则将自动对您进行身份验证，并打开 Web UI。

  如果票据过期了，需要首先使用凭证进行身份验证。但是，下次 IdM Web UI 将自动打开，而不会打开登录对话框。

  如果您看到错误消息 **Authentication with Kerberos failed**，请验证您的浏览器是否已针对 Kerberos 身份验证进行了配置。请参阅 为 Kerberos 身份验证配置浏览器。




## 6.5. 为 KERBEROS 身份验证配置外部系统

按照以下流程配置外部系统，以便身份管理(IdM)用户可以使用他们的 Kerberos 凭证从外部系统登录到 IdM。

当您的基础架构包含多个域或重叠域时，在外部系统上启用 Kerberos 身份验证非常有用。如果系统尚未通过 **ipa-client-install** 注册到任何 IdM 域，它也很有用。

要从不属于 IdM 域成员的系统启用对 IdM 的 Kerberos 身份验证，请在外部系统上定义特定于 IdM 的 Kerberos 配置文件。

**先决条件**

- **krb5-workstation** 软件包已安装在外部系统上。
  要查找是否安装了该软件包，请使用以下 CLI 命令：

```
# yum list installed krb5-workstation
Installed Packages
krb5-workstation.x86_64    1.16.1-19.el8    @BaseOS
```

**流程**

1. 将 **/etc/krb5.conf** 文件从 IdM 服务器复制到外部系统。例如：

```
# scp /etc/krb5.conf root@externalsystem.example.com:/etc/krb5_ipa.conf
```

> **警告**
>
> 不要覆盖外部系统上现有的 **krb5.conf** 文件。

2. 在外部系统上，将终端会话设置为使用复制的 IdM Kerberos 配置文件：

```
$ export KRB5_CONFIG=/etc/krb5_ipa.conf
```

-

**KRB5_CONFIG** 变量仅在退出之前暂时存在。要防止其丢失，请使用其他文件名导出变量。

3. 将 Kerberos 配置代码段从 **/etc/krb5.conf.d/** 目录复制到外部系统。
4. 在外部系统上配置浏览器，如 为 Kerberos 身份验证配置浏览器 中所述。

外部系统上的用户现在可以使用 **kinit** 工具对 IdM 服务器进行身份验证。

## 6.6. 活动目录用户的 WEB UI 登录

要为活动目录用户启用 Web UI 登录，请在 **Default Trust View** 中为每个活动目录用户定义一个 ID 覆盖。例如：

```
[admin@server ~]$ ipa idoverrideuser-add 'Default Trust View' ad_user@ad.example.com
```

**其它资源**

- 为活动目录用户使用 ID 视图

# 第 7 章 使用一次性密码登录到身份管理 WEB UI

可以通过多种方法保护对 IdM Web UI 的访问。最基本的一种是密码身份验证。

要提高密码身份验证的安全性，您可以添加第二个步骤，并需要自动生成的一次性密码(OTP)。最常见的用法是将与用户帐户连接的密码与由硬件或软件令牌生成的有时间限制的一次性密码结合起来。

以下章节可帮助到您：

- 了解 OTP 身份验证在 IdM 中的工作方式。
- 在 IdM 服务器上配置 OTP 身份验证。
- 在 IdM 中为 OTP 验证配置 RADIUS 服务器。
- 创建 OTP 令牌，并将它们与您电话中的 FreeOTP 应用程序同步。
- 使用用户密码和一次性密码的组合，向 IdM Web UI 进行身份验证。
- 在 Web UI 中重新同步令牌.
- 以 OTP 或 RADIUS 用户身份检索 IdM 票据授予票据

## 7.1. 先决条件

- 在 web 浏览器中访问 IdM Web UI

## 7.2. 身份管理中的一次性密码(OTP)身份验证

一次性密码可为您的身份验证安全性增加一步。身份验证使用您的密码 + 自动生成的一次性密码。

要生成一次性密码，您可以使用硬件或软件令牌。IdM 同时支持软件和硬件令牌。

身份管理支持以下两个标准的 OTP 机制：

- 基于 HMAC 的一次性密码(HOTP)算法是基于计数器的。HMAC 代表哈希消息身份验证代码。
- 基于时间的一次性密码(TOTP)算法是 HOTP 的扩展，来支持基于时间的移动因子。

**重要**

IdM 不支持活动目录信任用户的 OTP 登录。

## 7.3. 在 WEB UI 中启用一次性密码

身份管理(IdM)管理员可以全局或单独为 IdM 用户启用双因素身份验证(2FA)。用户在命令行上的常规密码后或者 Web UI 登录对话框中的专用字段输入一次性密码(OTP)，在这些密码之间没有空格。

启用 2FA 与强制使用它不同。如果您使用基于 LDAP 绑定的登录，IdM 用户仍可以只通过输入密码来进行身份验证。但是，如果您使用基于 **krb5** 的登录，则强制使用 2FA。在以后的发行版本中，红帽计划为管理员提供配置选项，以选择以下之一：

- 允许用户设置自己的令牌。在这种情况下，LDAP 绑定仍然不会强制执行 2FA，但基于 **krb5** 的登录会强制执行 2FA。

- 不允许用户设置自己的令牌。在这种情况下，在基于 LDAP 绑定和 **krb5** 的登录中会强制使用 2FA。

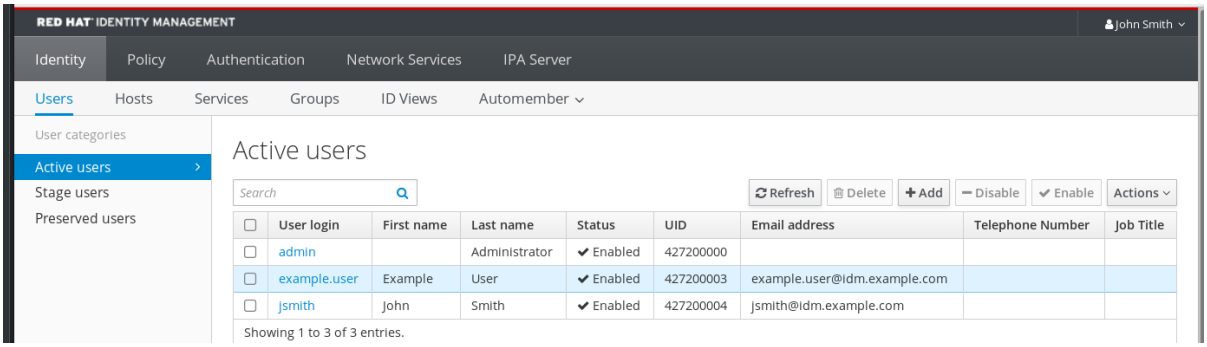
完成此流程，以使用 IdM Web UI 为单独的 **example.user** IdM 用户启用 2FA。

**先决条件**

- 管理特权

**流程**

1. 使用 IdM **admin** 权限登录到 IdM Web UI。
2. 打开 **Identity → Users → Active users**选项卡。




3. 选择 **example.user** 以打开用户设置。
4. 在 **User authentication types** 中，选择 **Two factor authentication (password + OTP)**。
5. 点击 **Save**。

此时，已为 IdM 用户启用了 OTP 身份验证。

现在，您或 **example.user** 必须向 **example.user** 帐户分配一个新的令牌 ID。

## 7.4. 在 IDM 中为 OTP 验证配置 RADIUS 服务器

要启用从专用一次性密码(OTP)解决方案到身份管理(IdM)原生 OTP 解决方案的大型部署的迁移，IdM 提供了为用户子集将 OTP 验证卸载到的第三方 RADIUS 服务器的方法。管理员创建了一组 RADIUS 代理，其中每个代理只能引用一个 RADIUS 服务器。如果需要寻址多个服务器，建议创建一个指向多个 RADIUS 服务器的虚拟 IP 解决方案。

例如，此类解决方案必须在 RHEL IdM 外部构建，例如，在 **keepalived** 守护进程的帮助下。然后，管理员将这些代理集中的一个分配给用户。只要用户分配了 RADIUS 代理集，IdM 就会绕过所有其他身份验证机制。

**注意**

IdM 不为第三方系统中的令牌提供任何令牌管理或同步支持。

完成流程，来为 OTP 验证配置 RADIUS 服务器，并将用户添加到代理服务器：

**先决条件**

- radius 用户验证方法已启用。详情请参阅 在 Web UI 中启用一次性密码。

**流程**

1. 添加一个 RADIUS 代理：

```
$ ipa radiusproxy-add proxy_name --secret secret
```

命令提示您插入所需的信息。

RADIUS 代理的配置需要在客户端和服务器之间使用通用 secret 来包装凭证。在 **--secret** 参数中指定此 secret。

2. 向添加的代理分配用户：

```
ipa user-mod radiususer --radius=proxy_name
```

3. 如果需要，配置要发送到 RADIUS 的用户名：

```
ipa user-mod radiususer --radius-username=radius_user
```

因此，RADIUS 代理服务器开始处理用户 OTP 身份验证。

当用户准备好迁移到 IdM 原生 OTP 系统时，您可以简单地删除用户的 RADIUS 代理分配。

### 7.4.1. 在较慢的网络中运行 RADIUS 服务器时更改 KDC 的超时值

在某些情况下，比如在较慢的网络中运行 RADIUS 代理，身份管理(IdM) Kerberos 分发中心(KDC)会在 RADIUS 服务器响应前关闭连接，因为在等待用户输入令牌过程中连接超时。

要更改 KDC 的超时设置：

1. 更改 **/var/kerberos/krb5kdc/kdc.conf** 文件中 **[otp]** 部分中 **timeout** 参数的值。例如，要将超时设置为 **120** 秒：

```
[otp]
DEFAULT = {
  timeout = 120
  ...
}
```

2. 重启 **krb5kdc** 服务：

```
# systemctl restart krb5kdc
```

**其它资源**

- 如何在 FIPS 模式下配置 FreeRADIUS 身份验证 知识库文章

## 7.5. 在 WEB UI 中添加 OTP 令牌

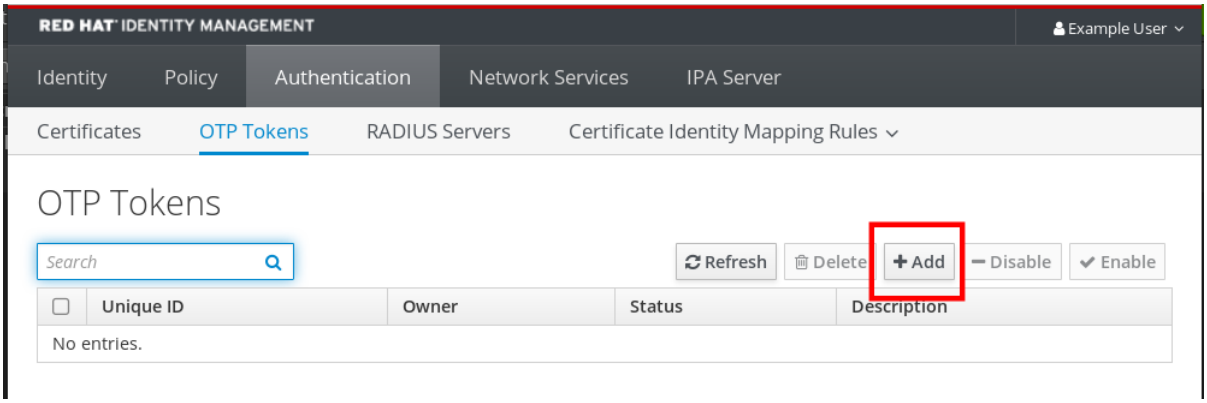
下面的章节帮助您将令牌添加到 IdM Web UI，以及您的软件令牌生成器中。

**先决条件**

- IdM 服务器上的活跃用户帐户。
- 管理员已在 IdM Web UI 中为特定用户帐户启用了 OTP。
- 生成 OTP 令牌的软件设备，如 FreeOTP。

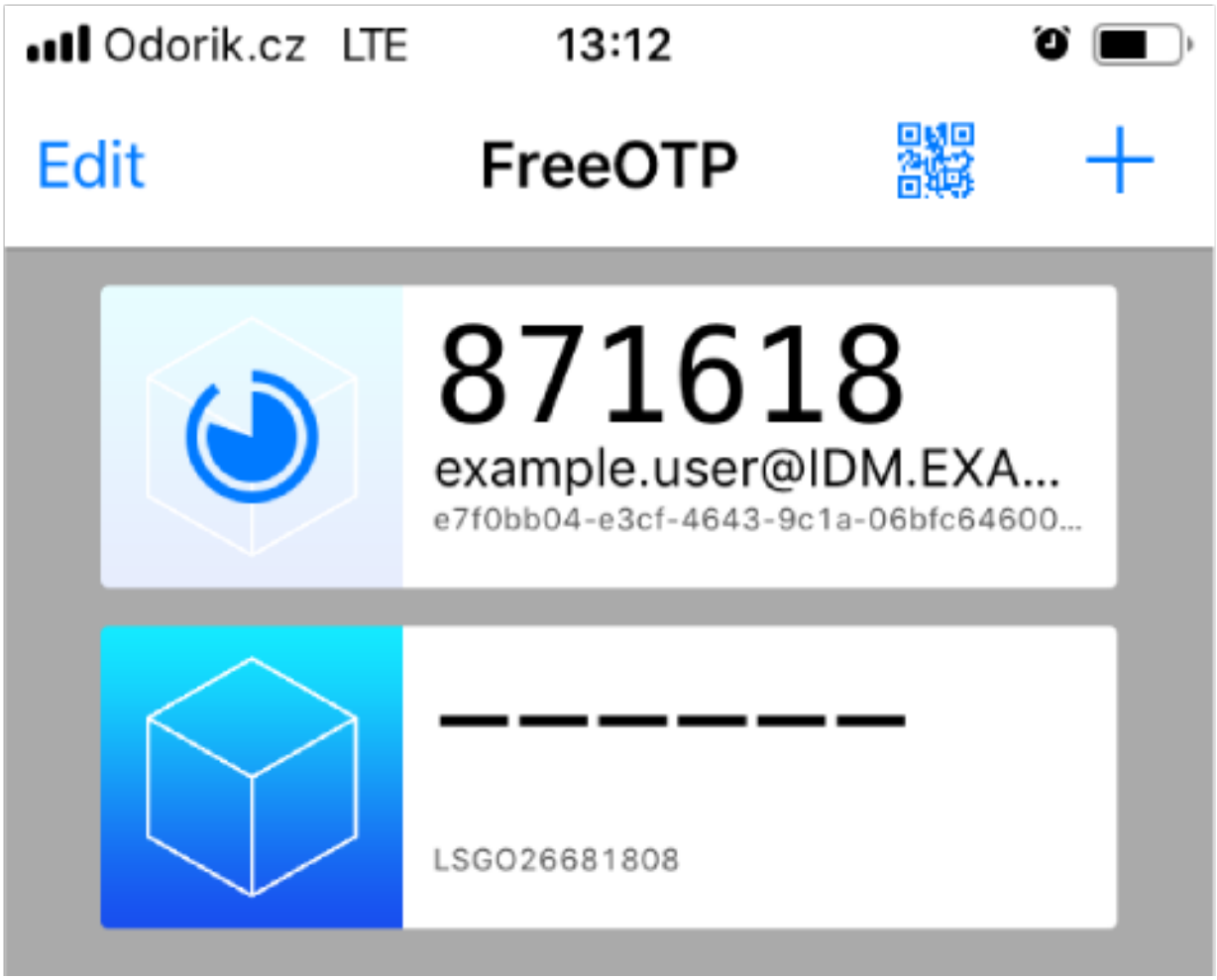
**流程**

1. 使用您的用户名和密码登录到 IdM Web UI。
2. 要在您的手机中创建令牌，请打开 **Authentication → OTP Tokens**选项卡。
3. 点击 **Add**。




4. 在 **Add OTP 令牌**对话框中，保留所有内容未填充，并点击 **Add**。
   在这个阶段，IdM 服务器在服务器上创建一个带有默认参数的令牌，并打开一个带有 QR 代码的页面。
5. 将 QR 代码复制到您的手机。
6. 单击 **OK** 来关闭 QR 代码。

现在，您可以生成一次性密码，并使用它们登录到 IdM Web UI。




## 7.6. 使用一次性密码登录到 WEB UI

按照以下流程，使用一次性密码(OTP)首次登录到 IdM Web UI。

**先决条件**

- OTP 配置在身份管理服务器上为用于 OTP 身份验证的用户帐户启用 OTP 配置。管理员和用户本身也可以启用 OTP。
  要启用 OTP 配置，请参阅 在 Web UI 中启用一次性密码。
- 生成 OTP 令牌的硬件或软件设备已配置。

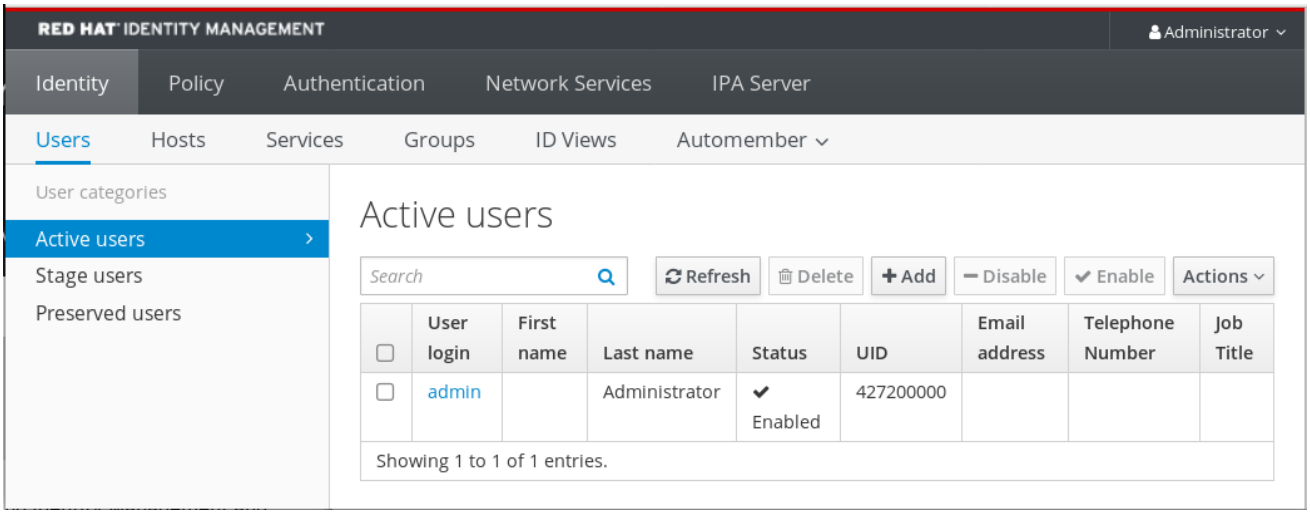
**流程**

1. 在身份管理登录屏幕中，输入您的用户名或 IdM 服务器管理员帐户的用户名。
2. 为上面输入的用户名添加密码。
3. 在您的设备上生成一次性密码。
4. 在密码后面输入一次性密码（不带空格）。
5. 点击 **Log in**。
   如果身份验证失败，请同步 OTP 令牌。

   如果您的 CA 使用自签名证书，则浏览器会发出警告。检查证书并接受安全例外以进行登录。

如果 IdM Web UI 没有打开，请验证您的身份管理服务器的 DNS 配置。

登录成功后，会出现 IdM Web UI。




## 7.7. 使用 WEB UI 同步 OTP 令牌

如果使用 OTP 登录（一次性密码）失败，OTP 令牌不会被正确同步。

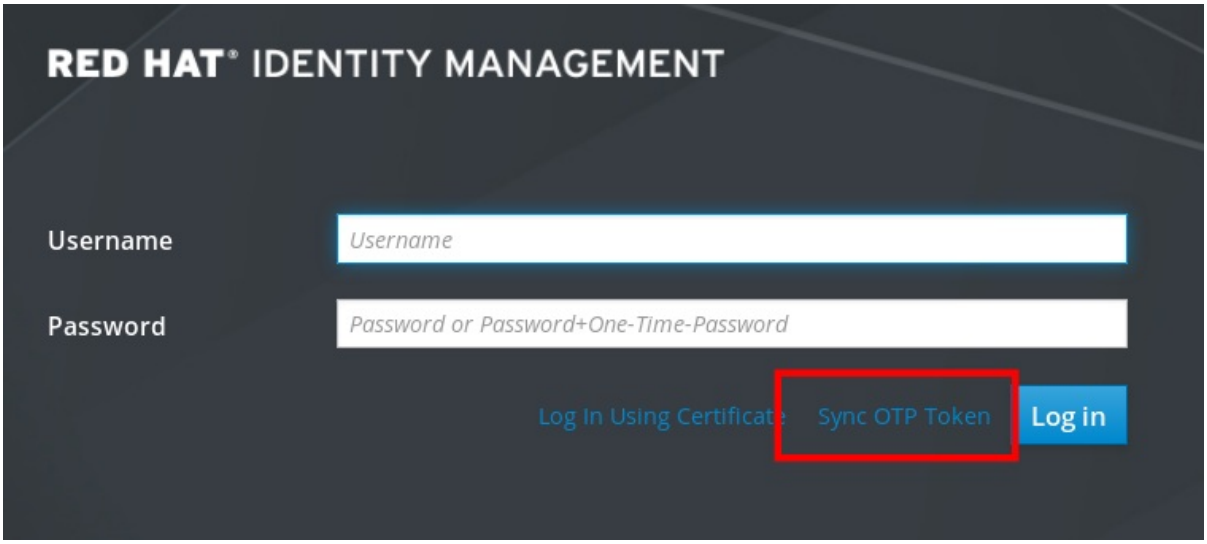
以下文本描述了令牌重新同步。

**先决条件**

- 登录屏幕已打开。
- 生成 OTP 令牌的设备已配置。

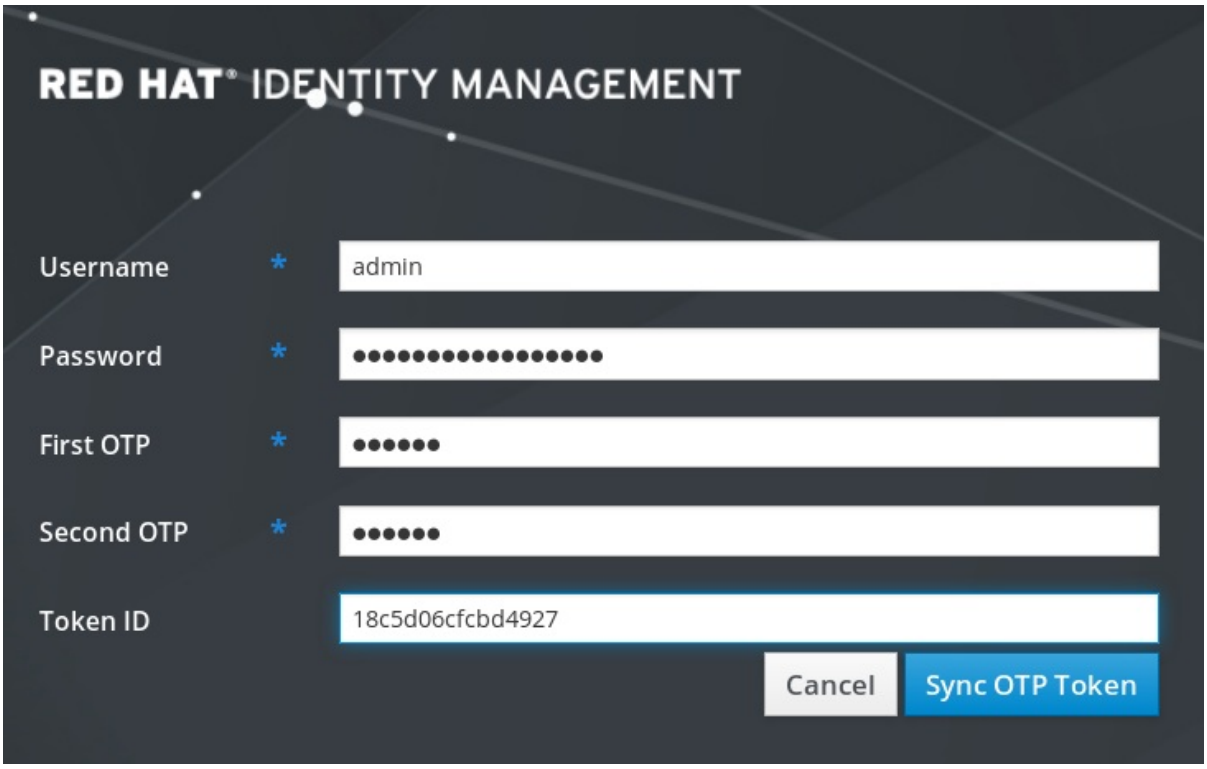
**流程**

1. 在 IdM Web UI 登录屏幕上，单击 **Sync OTP Token**。




2. 在登录屏幕中，输入您的用户名和身份管理密码。
3. 生成一次性密码，并将它输入到 **First OTP** 字段中。
4. 再生成一次性密码，并将它输入到 **Second OTP** 字段中。

5. （可选）输入令牌 ID。




6. 单击 **Sync OTP Token**。

同步成功后，您可以登录到 IdM 服务器。

## 7.8. 更改过期的密码

身份管理的管理员可以强制您在下一次登录时更改密码。这意味着，在更改密码之前，您无法成功登录到 IdM Web UI。

您第一次登录到 Web UI 时可能会出现密码过期。

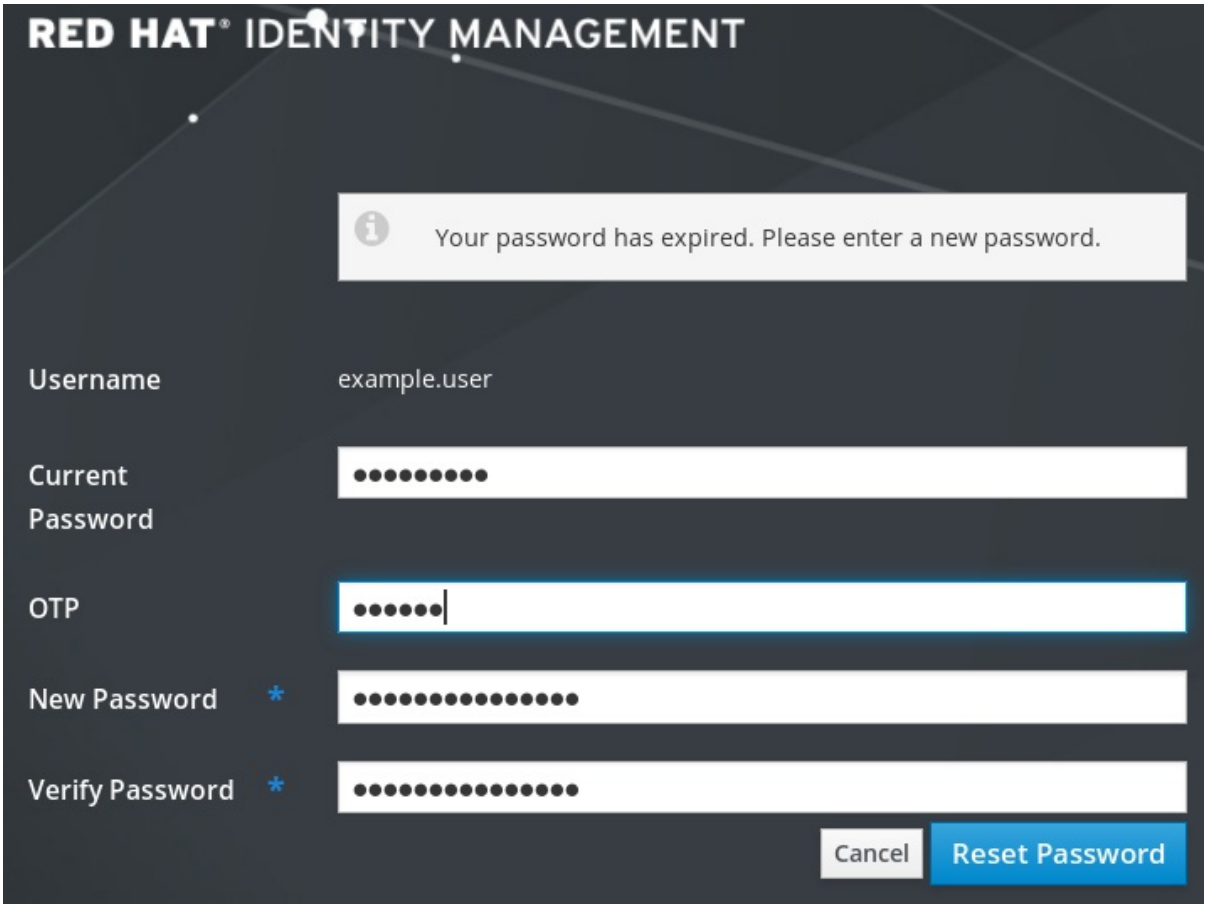
如果出现密码过期对话框，请按照流程中的说明操作。

**先决条件**

- 登录屏幕已打开。
- IdM 服务器的活动帐户。

**流程**

1. 在密码过期登录屏幕中，输入用户名。
2. 为上面输入的用户名添加密码。
3. 在 OTP 字段中，如果使用一次性密码身份验证，请生成一次性密码。
   如果您没有启用 OTP 身份验证，请将该字段留空。
4. 输入两次新密码进行验证。
5. 单击 **Reset Password**。




成功更改密码后，将显示常见的登录对话框。使用新密码登录。

## 7.9. 以 OTP 或 RADIUS 用户身份检索 IDM TICKET-GRANTING TICKET

要以 OTP 用户身份检索 Kerberos 票据授予票据(TGT)，请请求匿名 Kerberos 票据，并通过 Secure Tunneling (FAST)频道启用灵活的身份验证，以便在 Kerberos 客户端和 Kerberos 分发中心(KDC)之间提供安全连接。

**先决条件**

- 您的 IdM 客户端和服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 客户端和服务器使用 SSSD 2.7.0 或更高版本。
- 您已为所需用户帐户启用了 OTP。

**流程**

1. 运行以下命令来初始化凭证缓存：

```
[root@client ~]# kinit -n @IDM.EXAMPLE.COM -c FILE:armor.ccache
```

请注意，这个命令会创建 **armor.ccache** 文件，每当您请求新的 Kerberos 票据时，您需要指向该文件。

2. 运行以下命令来请求 Kerberos 票据：

```
[root@client ~]# kinit -T FILE:armor.ccache <username>@IDM.EXAMPLE.COM
Enter your OTP Token Value.
```

-

**验证**

- 显示您的 Kerberos 票据信息：

```
[root@client ~]# klist -C
Ticket cache: KCM:0:58420
Default principal: <username>@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
05/09/22 07:48:23  05/10/22 07:03:07  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: fast_avail(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = yes
08/17/2022 20:22:45  08/18/2022 20:22:43
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: pa_type(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = 141
```

**pa_type = 141** 表示 OTP/RADIUS 身份验证。

# 第 8 章 IDM 中 SSSD 身份验证故障排除

在 Identity Management（IdM）环境中的身份验证涉及许多组件：

**在 IdM 客户端中：**

- SSSD 服务。
- Name Services Switch (NSS)。
- 可插拔验证模块 (PAM)。

**在 IdM 服务器上：**

- SSSD 服务。
- IdM 目录服务器。
- IdM Kerberos 密钥分发中心 (KDC)。

**如果您要以 Active Directory (AD) 用户进行身份验证：**

- AD 域控制器上的目录服务器。
- AD 域控制器上的 Kerberos 服务器。

要验证用户，您必须使用 SSSD 服务执行以下功能：

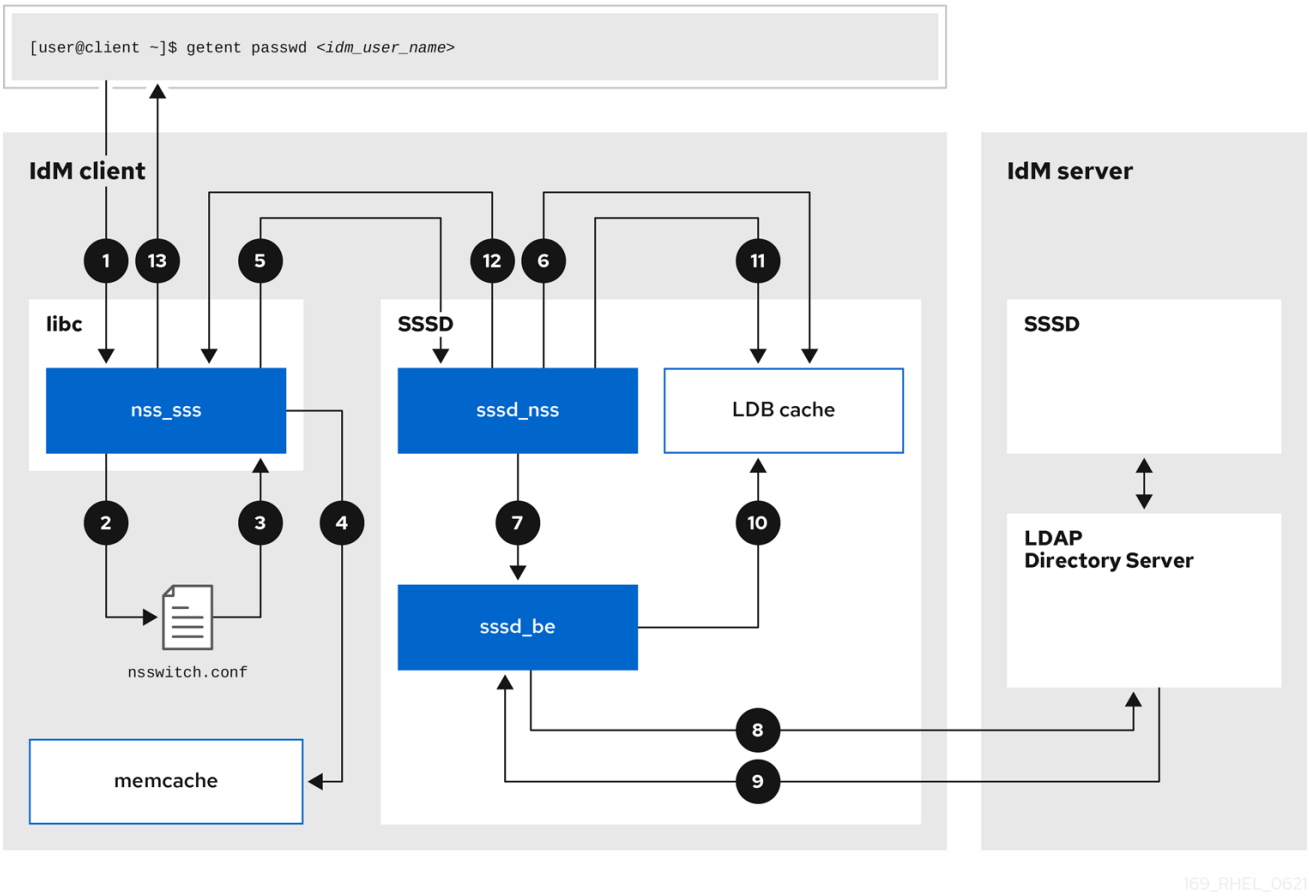
- 从身份验证服务器检索用户信息。
- 提示用户输入其凭据，将这些凭据传递到身份验证服务器，然后处理结果。

要了解更多有关 SSSD 服务和存储用户信息的服务器之间的信息流，以便可以排除您环境中身份验证尝试失败的问题，请参阅：

1. 使用 SSSD 获取 IdM 用户信息时的数据流
2. 使用 SSSD 获取 AD 用户信息时的数据流
3. 以 IdM 中的 SSSD 用户身份进行身份验证时的数据流
4. 缩小身份验证问题的范围
5. SSSD 日志文件和日志记录级别
6. 在 sssd.conf 文件中为 SSSD 启用详细日志记录
7. 使用 sssctl 命令为 SSSD 启用详细的日志记录
8. 从 SSSD 服务收集调试日志，对 IdM 服务器的身份验证问题进行故障排除
9. 从 SSSD 服务收集调试日志，以对 IdM 客户端的身份验证问题进行故障排除
10. 跟踪 SSSD 后端中的客户端请求
11. 使用日志分析器工具跟踪客户端请求

## 8.1. 使用 SSSD 获取 IDM 用户信息时的数据流

下图使用 **getent passwd *<idm_user_name>*** 命令在请求 IdM 用户信息的过程中简化 IdM 客户端和 IdM 服务器之间的信息流。



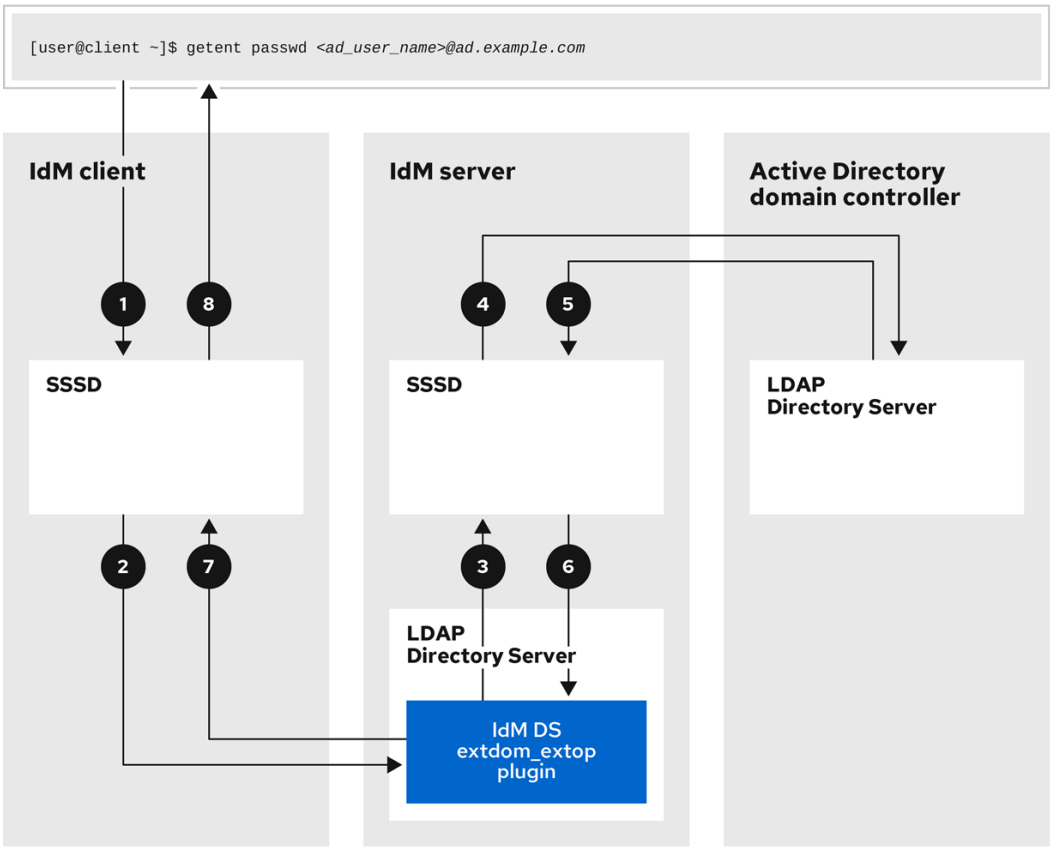

1. **getent** 命令会触发来自 **libc** 库的 **getpwnam** 调用。
2. **libc** 库引用 **/etc/nsswitch.conf** 配置文件来检查哪个服务负责提供用户信息，并发现 **SSSD** 服务的条目。
3. **libc** 库打开 **ss_sss** 模块。
4. nss_sss 模块检查内存映射缓存以获取用户信息。如果缓存中存在数据，则 **ss_sss** 模块会返回它。
5. 如果用户信息不在内存映射缓存中，则会将请求传递给 SSSD **sssd_nss** 响应程序进程。
6. SSSD 服务检查其缓存。如果缓存中存在数据并有效，**sssd_nss** 响应程序会从缓存中读取数据并将其返回到应用。
7. 如果缓存中没有数据或数据已过期，**sssd_nss** 响应器将查询相应的后端进程并等待回复。SSSD 服务在 IdM 环境中使用 IPA 后端，通过 **sssd.conf** 配置文件中的 **id_provider=ipa** 启用。
8. **sssd_be** 后端进程连接到 IdM 服务器，并从 IdM LDAP 目录服务器请求信息。
9. IdM 服务器上的 SSSD 后端响应 IdM 客户端上的 SSSD 后端进程。
10. 客户端上的 SSSD 后端将生成的数据存储在 SSSD 缓存中，并提醒已更新缓存的响应程序进程。

11. **sssd_nss** 前端响应器进程从 SSSD 缓存检索信息。
12. **sssd_nss** 响应器将用户信息发送到 **ss_sss** 响应者，以完成请求。
13. **libc** 库将用户信息返回到请求它的应用程序。

## 8.2. 使用 SSSD 获取 AD 用户信息时的数据流

如果您已在 IdM 环境和活动目录(AD)域之间建立了跨林信任，则检索 IdM 客户端的 AD 用户信息时的信息流与检索 IdM 用户信息时的信息流非常相似，只是多了一个联系 AD 用户数据库的步骤。

下图是当用户使用 **getent passwd *<ad_user_name@ad.example.com>*** 命令请求 AD 用户的信息时，信息流的一种简化。这个图并没有包括使用 SSSD 检索 IdM 用户信息时的数据流 中讨论的内部详细信息。它侧重于 IdM 客户端上的 SSSD 服务、IdM 服务器上的 SSSD 服务和 AD 域控制器上的 LDAP 数据库之间的通信。




1. IdM 客户端为 AD 用户信息查找其本地 SSSD 缓存。
2. 如果 IdM 客户端没有用户信息，或者信息是 stale，客户端上的 SSSD 服务会联系 IdM 服务器上的 **extdom_extop** 插件来执行 LDAP 扩展操作并请求信息。
3. IdM 服务器上的 SSSD 服务在其本地缓存中查找 AD 用户信息。
4. 如果 IdM 服务器在其 SSSD 缓存中没有用户信息，或者其信息为过时，它将执行 LDAP 搜索，以从 AD 域控制器请求用户信息。
5. IdM 服务器上的 SSSD 服务从 AD 域控制器接收 AD 用户信息，并将其存储在其缓存中。
6. **extdom_extop** 插件从 IdM 服务器上的 SSSD 服务接收信息，该服务完成 LDAP 扩展操作。

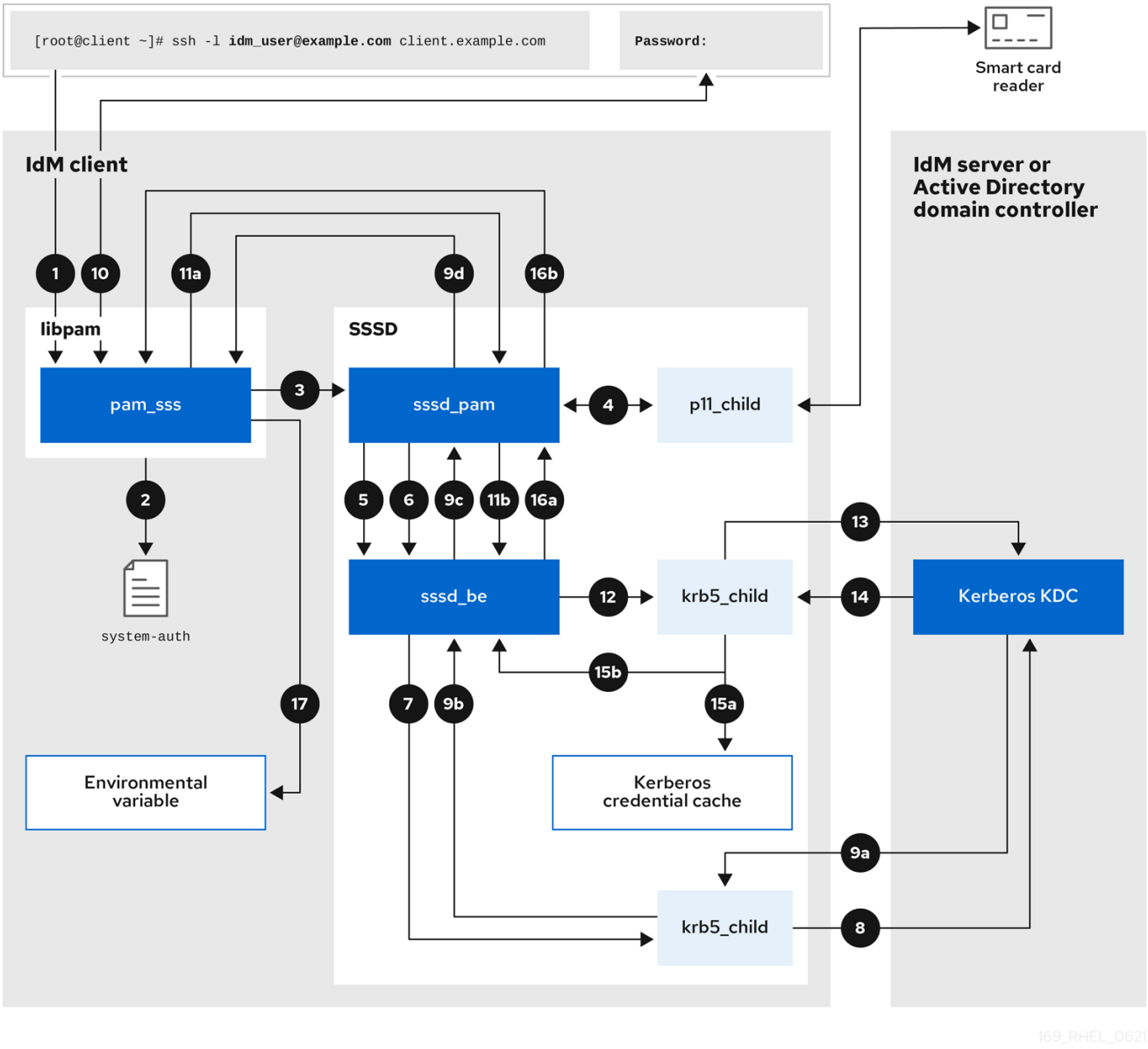
7. IdM 客户端上的 SSSD 服务从 LDAP 扩展操作接收 AD 用户信息。
8. IdM 客户端将 AD 用户信息存储在其 SSSD 缓存中，并将信息返回给请求它的应用程序。

## 8.3. 以 IDM 中的 SSSD 用户身份进行身份验证时的数据流

以 IdM 服务器或客户端中的用户身份进行身份验证涉及以下组件：

- 启动身份验证请求的服务，如 sshd 服务。
- 可插拔验证模块 (PAM) 库及其模块。
- SSSD 服务、其响应者和后端。
- 智能卡读取器（如果配置了智能卡验证）。
- 身份验证服务器：
  - IdM 用户通过 IdM Kerberos 密钥分发中心 (KDC) 进行身份验证。
  - Active Directory (AD) 用户通过 AD 域控制器 (DC) 进行身份验证。

下图是用户在尝试通过命令行上的 SSH 服务在本地登录主机期间需要进行身份验证时的简化信息流。




1. 使用 **ssh** 命令尝试身份验证会触发 **libpam** 库。

2. **libpam** 库引用 **/etc/pam.d/** 目录中与请求身份验证尝试的服务对应的 PAM 文件。在本例中，libpam 库涉及通过本地主机上的 SSH 服务进行身份验证，**libpam** 库检查 **/etc/pam.d/system-auth** 配置文件并发现 SSSD PAM 的 **pam_sss.so** 条目：

   ```
   auth    sufficient    pam_sss.so
   ```

3. 要确定哪些身份验证方法可用，**libpam** 库会打开 **pam_sss** 模块，并将 **SSS_PAM_PREAUTH** 请求发送到 SSSD 服务的 **sssd_pam** PAM 响应者。

4. 如果配置了智能卡验证，SSSD 服务会生成一个临时 **p11_child** 进程，以检查智能卡并从中检索证书。

5. 如果为用户配置了智能卡验证，**sssd_pam** 响应程序会尝试将智能卡中的证书与用户匹配。**sssd_pam** 响应器还搜索用户所属的组，因为组成员身份可能会影响访问控制。

6. **sssd_pam** 响应程序将 **SSS_PAM_PREAUTH** 请求发送到 **sssd_be** 后端响应程序，以查看服务器支持的身份验证方法，如密码或双因素身份验证。在 IdM 环境中，SSSD 服务使用 IPA 响应器，默认的身份验证方法是 Kerberos。在本例中，用户使用简单的 Kerberos 密码进行身份验

证。

7. **sssd_be** 响应器生成一个临时 **krb5_child** 进程。
8. **krb5_child** 进程联系 IdM 服务器上的 KDC，并检查可用的身份验证方法。
9. KDC 响应请求：
   a. **krb5_child** 进程评估回复，并将结果发回到 **sssd_be** 后端进程。
   b. **sssd_be** 后端进程会收到结果。
   c. **sssd_pam** 响应器会收到结果。
   d. **pam_sss** 模块会收到结果。
10. 如果为用户配置了密码身份验证，**pam_sss** 模块将提示用户输入其密码。如果配置了智能卡验证，**pam_sss** 模块会提示用户输入其智能卡 PIN。
11. 模块会发送带有用户名和密码的 **SSS_PAM_AUTHENTICATE** 请求，该请求经过以下操作：
    a. **sssd_pam** 响应器。
    b. **sssd_be** 后端进程。
12. **sssd_be** 进程生成一个临时 **krb5_child** 进程来联系 KDC。
13. **krb5_child** 进程尝试使用用户提供的用户名和密码从 KDC 检索 Kerberos Ticket Granting Ticket (TGT)。
14. **krb5_child** 进程接收身份验证尝试的结果。
15. **krb5_child** 进程：
    a. 将 TGT 存储到凭据缓存中。
    b. 将身份验证结果返回到 **sssd_be** 后端进程。
16. 身份验证结果从 **sssd_be** 进程传输到：
    a. **sssd_pam** 响应器。
    b. **pam_sss** 模块。
17. **pam_sss** 模块使用用户 TGT 的位置设置环境变量，以便其他应用可以引用它。

## 8.4. 缩小身份验证问题的范围

要成功验证用户，您必须能够使用 SSSD 服务从存储用户信息的数据库检索用户信息。以下流程描述了测试身份验证流程的不同组件的步骤，以便您可以在用户无法登录时缩小身份验证问题的范围。

**流程**

1. 验证 SSSD 服务及其进程是否正在运行。

```
[root@client ~]# pstree -a | grep sssd
  |-sssd -i --logger=files
```

```
|   |-sssd_be --domain implicit_files --uid 0 --gid 0 --logger=files
|   |-sssd_be --domain example.com --uid 0 --gid 0 --logger=files
|   |-sssd_ifp --uid 0 --gid 0 --logger=files
|   |-sssd_nss --uid 0 --gid 0 --logger=files
|   |-sssd_pac --uid 0 --gid 0 --logger=files
|   |-sssd_pam --uid 0 --gid 0 --logger=files
|   |-sssd_ssh --uid 0 --gid 0 --logger=files
|   `-sssd_sudo --uid 0 --gid 0 --logger=files
|-sssd_kcm --uid 0 --gid 0 --logger=files
```

2. 验证客户端可以通过 IP 地址联系用户数据库服务器。

```
[user@client ~]$ ping <IP_address_of_the_database_server>
```

如果此步骤失败，请检查您的网络和防火墙设置是否允许 IdM 客户端和服务器之间进行直接通信。请参阅使用和配置 firewalld。

3. 验证客户端可以通过完全限定的主机名发现并联系 IdM LDAP 服务器（适用于 IdM 用户）或 AD 域控制器（ AD 用户）。

```
[user@client ~]$ dig -t SRV _ldap._tcp.example.com @<name_server>
[user@client ~]$ ping <fully_qualified_host_name_of_the_server>
```

如果此步骤失败，请检查您的 Dynamic Name Service (DNS) 设置，包括 **/etc/resolv.conf** 文件。请参阅配置 DNS 服务器顺序。

**注意**

默认情况下，SSSD 服务会尝试通过 DNS 服务 (SRV) 记录自动发现 LDAP 服务器和 AD DC。另外，您可以通过在 **sssd.conf** 配置文件中设置以下选项，将 SSSD 服务限制为使用特定的服务器：

- **ipa_server = *<fully_qualified_host_name_of_the_server>***
- **ad_server = *<fully_qualified_host_name_of_the_server>***
- **ldap_uri = *<fully_qualified_host_name_of_the_server>***

如果使用这些选项，请验证您可以联系它们中列出的服务器。

4. 验证客户端是否可以对 LDAP 服务器进行身份验证，并使用 **ldapsearch** 命令检索用户信息。

   a. 如果您的 LDAP 服务器是 IdM 服务器，如 **server.example.com**，检索主机的 Kerberos 票据，并使用主机 Kerberos 主体进行身份验证数据库搜索：

```
[user@client ~]$ kinit -k 'host/client.example.com@EXAMPLE.COM'
[user@client ~]$ ldapsearch -LLL -Y GSSAPI -h server.example.com -b
“dc=example,dc=com” uid=<user_name>
```

   b. 如果您的 LDAP 服务器是 Active Directory (AD) 域控制器 (DC)，如 **server.ad.example.com**，请检索主机的 Kerberos 票据，并使用主机 Kerberos 主体执行数据库搜索：

```
[user@client ~]$ kinit -k 'CLIENT$@AD.EXAMPLE.COM'
[user@client ~]$ ldapsearch -LLL -Y GSSAPI -h server.ad.example.com -b
“dc=example,dc=com” sAMAccountname=<user_name>
```

c. 如果您的 LDAP 服务器是普通 LDAP 服务器，且您在 **sssd.conf** 文件中设置了 **ldap_default_bind_dn** 和 **ldap_default_authtok** 选项，请验证是同一个 **ldap_default_bind_dn** 帐户：

```
[user@client ~]$ ldapsearch -xLLL -D "cn=ldap_default_bind_dn_value" -W -h
ldapserver.example.com -b “dc=example,dc=com” uid=<user_name>
```

如果此步骤失败，请验证您的数据库设置是否允许您的主机搜索 LDAP 服务器。

5. 由于 SSSD 服务使用 Kerberos 加密，因此请以无法登录的用户身份获得 Kerberos 票据。

a. 如果您的 LDAP 服务器是 IdM 服务器：

```
[user@client ~]$ kinit <user_name>
```

b. 如果 LDAP 服务器数据库是 AD 服务器：

```
[user@client ~]$ kinit <user_name@AD.EXAMPLE.COM>
```

如果此步骤失败，请验证您的 Kerberos 服务器是否正常运行，所有服务器都已同步其时间，并且用户帐户未被锁定。

6. 验证您是否可以检索有关命令行的用户信息。

```
[user@client ~]$ getent passwd <user_name>
[user@client ~]$ id <user_name>
```

如果这一步失败，请验证客户端上的 SSSD 服务是否可以接收用户数据库的信息：

a. 查看 **/var/log/messages** 日志文件中的错误。

b. 在 SSSD 服务中启用详细的日志记录，收集调试日志，并查看日志以确定问题的根源。

c. *（可选）*创建一个红帽技术支持问题单，并提供您收集的故障排除信息。

7. 如果您被允许在主机上运行 **sudo**，请使用 **sssctl** 工具来验证用户是否被允许登录。

```
[user@client ~]$ sudo sssctl user-checks -a auth -s ssh <user_name>
```

如果这一步失败，请验证您的授权设置，如 PAM 配置、IdM HBAC 规则和 IdM RBAC 规则：

a. 确保用户的 UID 等于或大于 **UID_MIN**，它在 **/etc/login.defs** 文件中定义。

b. 查看 **/var/log/secure** 和 **/var/log/messages** 日志文件中的授权错误。

c. 在 SSSD 服务中启用详细的日志记录，收集调试日志，并查看日志以确定问题的根源。

d. *（可选）*创建一个红帽技术支持问题单，并提供您收集的故障排除信息。

**其它资源**

- 在 sssd.conf 文件中为 SSSD 启用详细日志记录
- 使用 sssctl 命令为 SSSD 启用详细的日志记录
- 从 SSSD 服务收集调试日志，对 IdM 服务器的身份验证问题进行故障排除
- 从 SSSD 服务收集调试日志，以对 IdM 客户端的身份验证问题进行故障排除

## 8.5. SSSD 日志文件和日志记录级别

每个 SSSD 服务都记录到 **/var/log/sssd/** 目录中自己的日志文件。对于 **example.com** IdM 域中的 IdM 服务器，其日志文件可能类似这样：

```
[root@server ~]# ls -l /var/log/sssd/
total 620
-rw-------. 1 root root      0 Mar 29 09:21 krb5_child.log
-rw-------. 1 root root  14324 Mar 29 09:50 ldap_child.log
-rw-------. 1 root root 212870 Mar 29 09:50 sssd_example.com.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_ifp.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_implicit_files.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd.log
-rw-------. 1 root root 219873 Mar 29 10:03 sssd_nss.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_pac.log
-rw-------. 1 root root  13105 Mar 29 09:21 sssd_pam.log
-rw-------. 1 root root   9390 Mar 29 09:21 sssd_ssh.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_sudo.log
```

### 8.5.1. SSSD 日志文件用途

**krb5_child.log**

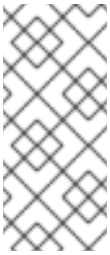
Kerberos 身份验证中涉及的短期帮助程序进程的日志文件。

**ldap_child.log**

与 LDAP 服务器通信的简短帮助程序进程的日志文件，涉及获取 Kerberos 票据。

**sssd_<example.com>.log**

对于 **sssd.conf** 文件中的每个域部分，SSSD 服务会将与 LDAP 服务器通信的信息记录到单独的日志文件中。例如，在名为 **example.com** 的 IdM 域环境中，SSSD 服务将其信息记录到名为 **sssd_example.com.log** 的文件中。如果主机直接与名为 **ad.example.com** 的 AD 域集成，信息将记录到名为 **sssd_ad.example.com.log** 的文件中。



**注意**

如果您有一个 IdM 环境以及与 AD 域的跨林信任，则有关 AD 域的信息仍会记录到 IdM 域的日志文件中。

类似地，如果主机直接集成到 AD 域，则任何子域的信息都会写入到主域的日志文件中。

**selinux_child.log**

用于检索和设置 SELinux 信息的短生命帮助器进程的日志文件。

**sssd.log**

SSSD 监控并与其响应器和后端进程通信的日志文件。

**sssd_ifp.log**

InfoPipe 响应器的日志文件，它提供了一个可通过系统总线访问的公共 D-Bus 接口。

**sssd_nss.log**

用于检索用户和组信息的 Name Services Switch (NSS) 响应器的日志文件。

**sssd_pac.log**

Microsoft Privilege Attribute 证书 (PAC) 响应器的日志文件，从 AD Kerberos 票据收集 PAC，并从 PAC 中生成 AD 用户的信息，从而避免直接从 AD 请求它。

**sssd_pam.log**

可插拔验证模块 (PAM) 响应器的日志文件。

**sssd_ssh.log**

SSH 响应器进程的日志文件。

### 8.5.2. SSSD 日志记录级别

设置一个 debug 级别后，也会启用它以下的所有 debug 级别。例如，把 debug 级别设置为 6 后，也会启用 debug 级别 0 到 5。

**表 8.1. SSSD 日志记录级别**

| 级别 | Description |
|---|---|
| 0 | **致命故障**。阻止 SSSD 服务启动或导致它终止的错误。这是 RHEL 8.3 及更早版本的默认调试日志级别。 |
| 1 | **关键故障**。错误没有导致 SSSD 服务被终止，但至少有一个主要功能无法正常工作。 |
| 2 | **严重故障**。这个错误声明特定请求或操作失败。这是 RHEL 8.4 及之后的版本的默认调试日志级别。 |
| 3 | **小故障**。在级别 2 中捕获的操作失败的错误。 |
| 4 | **配置设置**。 |
| 5 | **功能**数据。 |
| 6 | 跟踪**操作**功能的消息。 |
| 7 | 跟踪**内部控制**功能的消息。 |
| 8 | **功能内部**变量的内容。 |
| 9 | 极**低级别跟踪**信息。 |

## 8.6. 在 SSSD.CONF 文件中为 SSSD 启用详细日志记录

默认情况下，RHEL 8.4 及更新版本中的 SSSD 服务仅记录严重故障（调试级别 2），但不记录在对身份验证问题进行故障排除所需的详细级别。

要在 SSSD 服务重启过程中永久启用详细的日志记录，请在 **/etc/sssd/sssd.conf** 配置文件的每个部分添加 **debug_level=*<integer>*** 选项，其中 ***<integer>*** 值是一个 0 到 9 之间的数字。debug 级别 0 到 3 会记录大错误的日志，级别 8 和更高级别会提供大量详细的日志消息。**级别 6 是调试身份验证问题的一个良好起点。**

**先决条件**

- 您需要 root 密码来编辑 **sssd.conf** 配置文件并重新启动 SSSD 服务。

**流程**

1. 在文本编辑器中打开 **/etc/sssd/sssd.conf** 文件。
2. 将 **debug_level** 选项添加到文件的每个部分，并将 debug 级别设置为您选择的详细程度。

```
[domain/example.com]
debug_level = 6
id_provider = ipa
...

[sssd]
debug_level = 6
services = nss, pam, ifp, ssh, sudo
domains = example.com

[nss]
debug_level = 6

[pam]
debug_level = 6

[sudo]
debug_level = 6

[ssh]
debug_level = 6

[pac]
debug_level = 6

[ifp]
debug_level = 6
```

3. 保存并关闭 **sssd.conf** 文件。
4. 重启 SSSD 服务以加载新的配置设置。

```
[root@server ~]# systemctl restart sssd
```

**其它资源**

- SSSD 日志文件和日志记录级别

## 8.7. 使用 SSSCTL 命令为 SSSD 启用详细的日志记录

默认情况下，RHEL 8.4 及更新版本中的 SSSD 服务仅记录严重故障（调试级别 2），但不记录在对身份验证问题进行故障排除所需的详细级别。

您可以在命令行中使用 **sssctl debug-level *<integer>*** 命令更改 SSSD 服务的 debug 级别，其中 ***<integer>*** 是 0 到 9 之间的一个数字。debug 级别 0 到 3 会记录大错误的日志，级别 8 和更高级别会提供大量详细的日志消息。级别 6 是调试身份验证问题的一个良好起点。

**先决条件**

- 您需要 root 密码来运行 **sssctl** 命令。

**流程**

- 使用 sssctl debug-level 命令将所选的调试级别设置为您所需的详细程度。

```
[root@server ~]# sssctl debug-level 6
```

**其它资源**

- SSSD 日志文件和日志记录级别

## 8.8. 从 SSSD 服务收集调试日志，对 IDM 服务器的身份验证问题进行故障排除

如果您在尝试以 IdM 用户身份对 IdM 服务器进行身份验证时遇到问题，请在服务器上的 SSSD 服务中启用详细的调试日志，并收集尝试检索用户信息的日志。

**先决条件**

- 您需要 root 密码来运行 **sssctl** 命令并重新启动 SSSD 服务。

**流程**

1. 在 IdM 服务器上启用详细的 SSSD 调试日志。

```
[root@server ~]# sssctl debug-level 6
```

2. 对于遇到身份验证问题的用户，在 SSSD 缓存中使相关的对象无效，这样使您不会绕过 LDAP 服务器来从缓存的 SSSD 中获取信息。

```
[root@server ~]# sssctl cache-expire -u idmuser
```

3. 通过删除旧的 SSSD 日志来最大程度减少数据集的故障排除。

```
[root@server ~]# sssctl logs-remove
```

4. 尝试切换至遇到身份验证问题的用户，同时在尝试前后收集时间戳。这些时间戳进一步缩小了数据集的范围。

```
[root@server sssd]# date; su idmuser; date
Mon Mar 29 15:33:48 EDT 2021
su: user idmuser does not exist
```

```
Mon Mar 29 15:33:49 EDT 2021
```

5. *（可选）如果您不想继续收集详细的*SSSD 日志，请降低 debug 级别。

```
[root@server ~]# sssctl debug-level 2
```

6. 查看 SSSD 日志，了解失败请求的信息。例如，检查 **/var/log/sssd/sssd_example.com.log** 文件表明 SSSD 服务没有在 **cn=accounts,dc=example,dc=com** LDAP 子树中找到用户。这可能表示用户不存在，或者存在于其他位置。

```
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [dp_get_account_info_send] (0x0200):
Got request for [0x1][BE_REQ_USER][name=idmuser@example.com]
...
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sdap_get_generic_ext_step] (0x0400):
calling ldap_search_ext with [(&(uid=idmuser)(objectclass=posixAccount)(uid=)(&
(uidNumber=)(!(uidNumber=0))))][cn=accounts,dc=example,dc=com].
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sdap_get_generic_op_finished]
(0x0400): Search result: Success(0), no errmsg set
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sdap_search_user_process] (0x0400):
Search for users, returned 0 results.
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sysdb_search_by_name] (0x0400):
No such entry
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sysdb_delete_user] (0x0400): Error: 2
(No such file or directory)
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sysdb_search_by_name] (0x0400):
No such entry
(Mon Mar 29 15:33:49 2021) [sssd[be[example.com]]]
[ipa_id_get_account_info_orig_done] (0x0080): Object not found, ending request
```

7. 如果您无法确定导致身份验证问题的原因：

   a. 收集您最近生成的 SSSD 日志。

   ```
   [root@server ~]# sssctl logs-fetch sssd-logs-Mar29.tar
   ```

   b. 创建一个红帽技术支持问题单并提供：

      i. SSSD 日志：**sssd-logs-Mar29.tar**

      ii. 与日志对应的请求的控制台输出，包括时间戳和用户名：

      ```
      [root@server sssd]# date; id idmuser; date
      Mon Mar 29 15:33:48 EDT 2021
      id: ‘idmuser’: no such user
      Mon Mar 29 15:33:49 EDT 2021
      ```

## 8.9. 从 SSSD 服务收集调试日志，以对 IDM 客户端的身份验证问题进行故障排除

如果您在尝试以 IdM 用户身份向 IdM 客户端进行身份验证时遇到问题，请验证您是否可以检索有关 IdM 服务器的用户信息。如果您无法检索有关 IdM 服务器的用户信息，您将无法在 IdM 客户端上检索它（从 IdM 服务器检索信息）。

确认身份验证问题不源自 IdM 服务器后，从 IdM 服务器和 IdM 客户端收集 SSSD 调试日志。

**先决条件**

- 用户仅在 IdM 客户端而不是 IdM 服务器中存在身份验证问题。
- 您需要 root 密码来运行 **sssctl** 命令并重新启动 SSSD 服务。

**流程**

1. **在客户端上：** 在文本编辑器中打开 **/etc/sssd/sssd.conf** 文件。
2. **在客户端：** 将 **ipa_server** 选项添加到文件的 **[domain]** 部分，并将其设置为 IdM 服务器。这可避免 IdM 客户端自动发现其他 IdM 服务器，从而将此测试限制为一个客户端和一个服务器。

```
[domain/example.com]
ipa_server = server.example.com
...
```

3. **在客户端上：** 保存并关闭 **sssd.conf** 文件。
4. **在客户端上：**重启 SSSD 服务以加载配置更改。

```
[root@client ~]# systemctl restart sssd
```

5. **在服务器和客户端上：**启用详细的 SSSD 调试日志。

```
[root@server ~]# sssctl debug-level 6
```

```
[root@client ~]# sssctl debug-level 6
```

6. **在服务器和客户端中：**为遇到身份验证问题的用户验证 SSSD 缓存中的对象，因此您不用绕过 LDAP 数据库，并检索 SSSD 信息已经缓存。

```
[root@server ~]# sssctl cache-expire -u idmuser
```

```
[root@client ~]# sssctl cache-expire -u idmuser
```

7. **在服务器和客户端上：**通过删除旧的 SSSD 日志来最小化 dataset 故障排除。

```
[root@server ~]# sssctl logs-remove
```

```
[root@server ~]# sssctl logs-remove
```

8. **在客户端上：**尝试切换至遇到身份验证问题的用户，同时在尝试前后收集时间戳。这些时间戳进一步缩小了数据集的范围。

```
[root@client sssd]# date; su idmuser; date
Mon Mar 29 16:20:13 EDT 2021
su: user idmuser does not exist
Mon Mar 29 16:20:14 EDT 2021
```

9. *（可选）***在服务器和客户端上：**如果您不想继续收集详细的 SSSD 日志，请降低 debug 级别。

```
[root@server ~]# sssctl debug-level 0
```

```
[root@client ~]# sssctl debug-level 0
```

10. **服务器和客户端：**查看 SSSD 日志以获取有关失败请求的信息。
    a. 在客户端日志中查看来自客户端的请求。
    b. 在服务器日志中查看来自客户端的请求。
    c. 在服务器日志中检查请求的结果。
    d. 查看客户端收到来自服务器的请求结果的结果。
11. 如果您无法确定导致身份验证问题的原因：
    a. 收集您最近在 IdM 服务器和 IdM 客户端中生成的 SSSD 日志。根据主机名或角色标记它们。

```
[root@server ~]# sssctl logs-fetch sssd-logs-server-Mar29.tar
```

```
[root@client ~]# sssctl logs-fetch sssd-logs-client-Mar29.tar
```

    b. 创建一个红帽技术支持问题单并提供：
        i. SSSD 调试日志：
            A. 来自服务器的 **sssd-logs-server-Mar29.tar**。
            B. 来自客户端的 **sssd-logs-client-Mar29.tar**
        ii. 与日志对应的请求的控制台输出，包括时间戳和用户名：

```
[root@client sssd]# date; su idmuser; date
Mon Mar 29 16:20:13 EDT 2021
su: user idmuser does not exist
Mon Mar 29 16:20:14 EDT 2021
```

## 8.10. 跟踪 SSSD 后端中的客户端请求

SSSD 以异步方式处理请求，并将来自不同请求的消息添加到同一日志文件中，您可以使用唯一的请求标识符和客户端 ID 来在后端日志中跟踪客户端请求。唯一的请求标识符以 **RID#<integer>** 形式添加到调试日志中，客户端 ID 的格式为 **[CID #<integer>]**。这可让您隔离与单个请求相关的日志，您可以跨多个 SSSD 组件的日志文件从头到尾跟踪请求。

**先决条件**

- 您已启用了调试日志，并且已从 IdM 客户端提交了请求。
- 您必须具有 root 特权才能显示 SSSD 日志文件的内容。

**流程**

1. 要查看 SSSD 日志文件，请使用 **less** 工具打开日志文件。例如，查看 **/var/log/sssd/sssd_example.com.log** ：

```
[root@server ~]# less /var/log/sssd/sssd_example.com.log
```

2. 查看 SSSD 日志，以获取有关客户端请求的信息。

```
(2021-07-26 18:26:37): [be[testidm.com]] [dp_req_destructor] (0x0400): [RID#3] Number of
active DP request: 0
(2021-07-26 18:26:37): [be[testidm.com]] [dp_req_reply_std] (0x1000): [RID#3] DP Request
AccountDomain #3: Returning [Internal Error]: 3,1432158301,GetAccountDomain() not
supported
(2021-07-26 18:26:37): [be[testidm.com]] [dp_attach_req] (0x0400): [RID#4] DP Request
Account #4: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-07-26 18:26:37): [be[testidm.com]] [dp_attach_req] (0x0400): [RID#4] Number of
active DP request: 1
```

SSSD 日志文件中的这个示例输出显示了两个不同的请求的唯一标识符 **RID#3** 和 **RID#4**。

但是，对 SSSD 客户端接口的单一客户端请求通常会在后端触发多个请求，因此客户端请求和后端中的请求之间不是 1 到 1 的对应关系。虽然后端中的多个请求有不同的 RID 号，但每个初始后端请求都包括唯一的客户端 ID，以便管理员可以跟踪单个客户端请求的多个 RID 号。

以下示例显示了一个客户端请求 **[sssd.nss CID #1]** 和多个在后端生成的请求，**[RID#5]** 到 **[RID#13]**:

```
(2021-10-29 13:24:16): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#5] DP Request [Account #5]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:16): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#6] DP Request [AccountDomain
#6]: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:16): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#7] DP Request [Account #7]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#8] DP Request [Initgroups #8]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#9] DP Request [Account #9]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#10] DP Request [Account #10]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#11] DP Request [Account #11]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#12] DP Request [Account #12]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#13] DP Request [Account #13]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
```

## 8.11. 使用日志分析器工具跟踪客户端请求

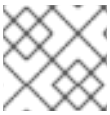
系统安全服务守护进程(SSSD)包含一个日志解析工具，可用于跟踪跨多个 SSSD 组件的日志文件的从头到尾的请求。

### 8.11.1. 日志分析器工具是如何工作的

使用日志解析工具，您可以跟踪跨多个 SSSD 组件的日志文件的从头到尾的 SSSD 请求。您可以使用 **sssctl analyze** 命令运行分析器工具。

日志分析器工具可帮助您排除 SSSD 中的 NSS 和 PAM 问题，并更容易查看 SSSD 调试日志。您只能提取和打印与跨多个 SSSD 进程的某些客户端请求有关的 SSSD 日志。

SSSD 跟踪用户和组身份信息(**id**,**getent**)，与用户身份验证（**su**、**ssh**）分开进行。NSS 响应器中的客户端 ID(CID)与 PAM 响应器中的 CID 无关，在分析 NSS 和 PAM 请求时会看到重叠的数字。在 **sssctl analyze** 命令中使用 **--pam** 选项来查看 PAM 请求。



**注意**

从 SSSD 内存缓存返回的请求不会被记录，并且日志分析器工具无法跟踪。

**其它资源**

- **sudo sssctl analyze request --help**
- **sudo sssctl analyze --help**
- **sssd.conf** 手册页
- **sssctl** 手册页

## 8.11.2. 运行日志分析器工具

按照以下流程，使用日志分析器工具跟踪 SSSD 中的客户端请求。

**先决条件**

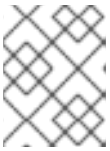
- 您必须在 [$responder] 部分和 **/etc/sssd/sssd.conf** 文件的 [domain/$domain] 部分中将 **debug_level** 至少设为 7，以启用日志解析功能。
- 要分析的日志必须来自使用 **libtevent** 链 ID 支持构建的 SSSD 的兼容版本，它是 RHEL 8.5 及之后版本中的 SSSD。

**流程**

1. 以 **列表** 模式运行日志分析器工具，来确定您在跟踪的请求的客户端 ID，添加 **-v** 选项以显示详细输出：

```
# sssctl analyze request list -v
```

将显示最近向 SSSD 发出的客户端请求的详细列表。



**注意**

如果分析 PAM 请求，请运行带有 **--pam** 选项的 **sssctl analyze request list** 命令。

2. 运行带有 **show [unique client ID]** 选项的日志分析器工具，来显示与指定客户端 ID 号相关的日志：

```
# sssctl analyze request show 20
```

3. 如果需要，您可以针对日志文件运行日志分析器工具，例如：

```
# sssctl analyze request --logdir=/tmp/var/log/sssd
```

**其它资源**

- **sssctl analyze request list --help**
- **sssctl analyze request show --help**
- **sssctl** 手册页。

## 8.12. 其它资源

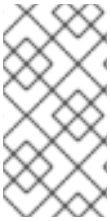
- 常规的 SSSD 调试流程

# 第 9 章 准备您的环境以使用 ANSIBLE PLAYBOOK 管理 IDM

作为管理身份管理 (IdM) 的系统管理员，在使用 Red Hat Ansible Engine 时，最好执行以下操作：

- 在您的主目录中保留一个专用于 Ansible playbook 的子目录，如 **~/MyPlaybooks**。
- 将示例 Ansible playbook 从 **/usr/share/doc/ansible-freeipa/*** 和 **/usr/share/doc/rhel-system-roles/*** 目录以及它们的子目录复制到 **~/MyPlaybooks** 目录中并进行调整。
- 将清单文件包含在 **~/MyPlaybooks** 目录中。

使用这个方法，您可以在一个位置找到所有 playbook。



**注意**

您可以在不调用受管节点上的 **root** 权限的情况下运行 **ansible-freeipa** playbook。例外包括使用 **ipaserver**、**ipareplica**、**ipaclient**、**ipasmartcard_server**、**ipasmartcard_client** 和 **ipabackup ansible-freeipa** 角色的 playbook。这些角色需要具有目录和 **dnf** 软件包管理器的特权访问权限。

Red Hat Enterprise Linux IdM 文档中的 playbook 假设以下 安全配置 ：

- IdM **admin** 是受管节点上的远程 Ansible 用户。
- 您可以将 IdM **admin** 密码加密存储在 Ansible 库中。
- 您已将保护 Ansible 库的密码放置在密码文件中。
- 您阻止对 vault 密码文件的访问权限，但您的本地 ansible 用户除外。
- 您定期删除并重新创建 vault 密码文件。

还要考虑 其他安全配置。

## 9.1. 准备控制节点和受管节点以使用 ANSIBLE PLAYBOOK 管理 IDM

按照以下流程创建 **~/MyPlaybooks** 目录并进行配置，以便您可以使用它来存储和运行 Ansible playbook。

**先决条件**

- 您已在受管节点上安装了 IdM 服务器 *server.idm.example.com* 和 *replica.idm.example.com*。
- 您已配置了 DNS 和网络，以便您可以直接从控制节点登录到受管节点 *server.idm.example.com* 和 *replica.idm.example.com*。
- 您需要知道 IdM **admin** 密码。

**流程**

1. 进入 **~/MyPlaybooks/** 目录：

```
$ cd ~/MyPlaybooks
```

2. 使用以下内容创建 **~/MyPlaybooks/ansible.cfg** 文件：

```
[defaults]
inventory = /home/your_username/MyPlaybooks/inventory
remote_user = admin
```

3. 使用以下内容创建 **~/MyPlaybooks/inventory** 文件：

```
[eu]
server.idm.example.com

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

此配置定义了两个主机组，即 **eu** 和 **us**，用于这些位置中的主机。此外，此配置定义了 **ipaserver** 主机组，它包含来自 **eu** 和 **us** 组的所有主机。

4. [可选] 创建 SSH 公钥和私钥。要在测试环境中简化访问，请不要在私钥中设置密码：

```
$ ssh-keygen
```

5. 将 SSH 公钥复制到每个受管节点上的 IdM **admin** 帐户：

```
$ ssh-copy-id admin@server.idm.example.com
$ ssh-copy-id admin@replica.idm.example.com
```

这些命令要求您输入 IdM **admin** 密码。

6. 创建包含 vault 密码的 **password_file** 文件：

```
redhat
```

7. 更改权限以修改该文件：

```
$ chmod 0600 password_file
```

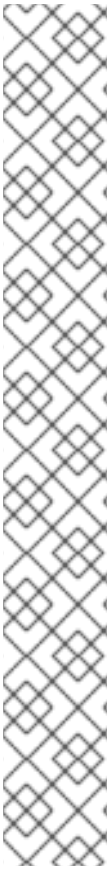
8. 创建一个 **secret.yml** Ansible 库来存储 IdM **admin** 密码：

   a. 配置 **password_file** 以存储 vault 密码：

   ```
   $ ansible-vault create --vault-password-file=password_file secret.yml
   ```

   b. 出现提示时，输入 **secret.yml** 文件的内容：

   ```
   ipaadmin_password: Secret123
   ```



**注意**

要在 playbook 中使用加密的 **ipaadmin_password**，您必须使用 **vars_file** 指令。例如，删除 IdM 用户的简单 playbook 可以如下所示：

```
---
- name: Playbook to handle users
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Delete user robot
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: robot
      state: absent
```

在执行 playbook 时，通过添加 **--vault-password-file=*password_file*选项，指示 Ansible 使用 vault 密码来解密 ipaadmin_password _password**。例如：

```
ansible-playbook -i inventory --vault-password-file=password_file del-user.yml
```

**警告**

为安全起见，删除每个会话末尾的 vault 密码文件，并在每个新会话开始时重复步骤 7-9。

**其它资源**

- 提供 ansible-freeipa playbook 所需的凭证的不同方法
- 使用 Ansible playbook 来安装身份管理服务器
- 如何构建清单

## 9.2. 提供 ANSIBLE-FREEIPA PLAYBOOK 所需的凭证的不同方法

不同的方法都有一些优点和缺点，为运行使用 **ansible-freeipa** 角色和模块的 playbook 提供所需的凭证。

### 在 playbook 中以纯文本形式存储密码

**优点** ：

- 运行 playbook 时，不会显示所有。
- 易于实施。

**缺陷** ：

- 有权访问该文件的人都可以读取密码。设置错误的权限和共享文件（例如在内部或外部存储库中）可能会破坏安全性。
- 高维护工作：如果更改了密码，则需要在所有 playbook 中进行更改。

### 执行 playbook 时以交互方式输入密码

**优点** ：

- 无人可以窃取密码，因为它不在任何位置存储。
- 您可以轻松更新密码。
- 易于实施。

**缺陷** ：

- 如果您在脚本中使用 Ansible playbook，以交互方式输入密码的要求可能比较不方便。

### 将密码存储在 Ansible vault 中，并将 vault 密码存储在一个文件中：

**优点** ：

- 用户密码以加密方式存储。
- 您可以通过创建新的 Ansible vault 来轻松更新用户密码。
- 您可以使用 **ansible-vault rekey --new-vault-password-file=NEW_VAULT_PASSWORD_FILE secret.yml** 命令来轻松更新保护 ansible vault 的密码文件。
- 如果您在脚本中使用 Ansible playbook，则不必以交互方式输入密码。

**缺陷** ：

- 务必要确保包含敏感纯文本密码的文件通过文件权限和其他安全措施进行保护。

### 将密码存储在 Ansible 库中，并以交互方式输入 vault 密码

**优点** ：

- 用户密码以加密方式存储。
- 无人可以窃取 vault 密码，因为它不在任何位置存储。
- 您可以通过创建新的 Ansible vault 来轻松更新用户密码。
- 您还可以使用 **ansible-vault rekey *file_name*** 命令，轻松更新 vault 密码。

**缺陷** ：

- 如果您在脚本中使用 Ansible playbook，则需要以交互方式输入 vault 密码。

### 其它资源

- 准备控制节点和受管节点以使用 Ansible playbook 管理 IdM
- 什么是 Zero 信任？
- 使用 Ansible vault 保护敏感数据

# 第 10 章 使用 ANSIBLE PLAYBOOK 配置全局 IDM 设置

使用 Ansible **config** 模块，您可以检索和设置 Identity Management (IdM) 的全局配置参数。

- 使用 Ansible playbook 检索 IdM 配置
- 使用 Ansible playbook 配置 IdM CA 续订服务器
- 使用 Ansible playbook 为 IdM 用户配置默认 shell
- 使用 Ansible 为 IdM 域配置 NETBIOS 名称
- 使用 Ansible 确保 IdM 用户和组有 SID

## 10.1. 使用 ANSIBLE PLAYBOOK 检索 IDM 配置

以下流程描述了如何使用 Ansible playbook 来检索有关当前全局 IdM 配置的信息。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible 库存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 打开 **/usr/share/doc/ansible-freeipa/playbooks/config/retrieve-config.yml** Ansible playbook 文件进行编辑：

```
---
- name: Playbook to handle global IdM configuration
  hosts: ipaserver
  become: no
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Query IPA global configuration
    ipaconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
    register: serverconfig
```

```
- debug:
    msg: "{{ serverconfig }}"
```

2. 通过更改以下内容来调整文件：
   - IdM 管理员的密码。
   - 其他值（如有必要）。
3. 保存该文件。
4. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 **secret.yml** 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/config/retrieve-config.yml
[...]
TASK [debug]
ok: [server.idm.example.com] => {
    "msg": {
        "ansible_facts": {
            "discovered_interpreter_
        },
        "changed": false,
        "config": {
            "ca_renewal_master_server": "server.idm.example.com",
            "configstring": [
                "AllowNThash",
                "KDC:Disable Last Success"
            ],
            "defaultgroup": "ipausers",
            "defaultshell": "/bin/bash",
            "emaildomain": "idm.example.com",
            "enable_migration": false,
            "groupsearch": [
                "cn",
                "description"
            ],
            "homedirectory": "/home",
            "maxhostname": "64",
            "maxusername": "64",
            "pac_type": [
                "MS-PAC",
                "nfs:NONE"
            ],
            "pwdexpnotify": "4",
            "searchrecordslimit": "100",
            "searchtimelimit": "2",
            "selinuxusermapdefault": "unconfined_u:s0-s0:c0.c1023",
            "selinuxusermaporder": [
                "guest_u:s0$xguest_u:s0$user_
            ],
            "usersearch": [
                "uid",
```

```
                "givenname",
                "sn",
                "telephonenumber",
                "ou",
                "title"
            ]
        },
        "failed": false
    }
}
```

## 10.2. 使用 ANSIBLE PLAYBOOK 配置 IDM CA 续订服务器

在使用嵌入式证书颁发机构 (CA) 的 Identity Management (IdM) 部署中，CA 续订服务器维护并更新 IdM 系统证书。它确保了强大的 IdM 部署。

有关 IdM CA 续订服务器角色的详情，请参阅 使用 IdM CA 续订服务器。

以下流程描述了如何使用 Ansible playbook 配置 IdM CA 续订服务器。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible 库存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 可选：识别当前 IdM CA 续订服务器：

```
$ ipa config-show | grep 'CA renewal'
  IPA CA renewal master: server.idm.example.com
```

2. 创建一个清单文件，如 **inventory.file**，并在该文件中定义 **ipaserver** ：

```
[ipaserver]
server.idm.example.com
```

3. 打开 **/usr/share/doc/ansible-freeipa/playbooks/config/set-ca-renewal-master-server.yml** Ansible playbook 文件进行编辑：

```
---
- name: Playbook to handle global DNS configuration
```

```
hosts: ipaserver
become: no
gather_facts: no
vars_files:
- /home/user_name/MyPlaybooks/secret.yml

tasks:
- name: set ca_renewal_master_server
  ipaconfig:
    ipaadmin_password: "{{ ipaadmin_password }}"
    ca_renewal_master_server: carenewal.idm.example.com
```

4. 通过更改调整文件：
   - **ipaadmin_password** 变量设置的 IdM 管理员密码。
   - **ca_renewal_master_server** 变量所设置的 CA 续订服务器的名称。
5. 保存该文件。
6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 **secret.yml** 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/config/set-ca-renewal-master-server.yml
```

**验证步骤**

您可以验证 CA 续订服务器是否已更改：

1. 以 IdM 管理员身份登录到 **ipaserver** ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 请求 IdM CA 续订服务器的身份：

```
$ ipa config-show | grep 'CA renewal'
IPA CA renewal master:  carenewal.idm.example.com
```

输出显示 **watchnewal.idm.example.com** 服务器是新的 CA 续订服务器。

## 10.3. 使用 ANSIBLE PLAYBOOK 为 IDM 用户配置默认 SHELL

shell 是一个接受和解释命令的程序。Red Hat Enterprise Linux (RHEL) 中提供了多个 shell，如 **bash**、**sh**、**ksh**、**zsh**、**fish** 等。**Bash** 或 **/bin/bash** 是大多数 Linux 系统中常用的 shell，它通常是 RHEL 上用户帐户的默认 shell。

以下流程描述了如何使用 Ansible playbook 将 **sh** （替代 shell）配置为 IdM 用户的默认 shell。

**先决条件**

- 您知道 IdM 管理员密码。

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible 库存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 可选： 使用 **retrieve-config.yml** Ansible playbook 来识别 IdM 用户的当前 shell。详情请参阅 使用 Ansible playbook 检索 IdM 配置。
2. 创建一个清单文件，如 **inventory.file**，并在该文件中定义 **ipaserver** ：

```
[ipaserver]
server.idm.example.com
```

3. 打开 **/usr/share/doc/ansible-freeipa/playbooks/config/ensure-config-options-are-set.yml** Ansible playbook 文件进行编辑：

```
---
- name: Playbook to ensure some config options are set
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  # Set defaultlogin and maxusername
  - ipaconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      defaultshell: /bin/bash
      maxusername: 64
```

4. 通过更改以下内容来调整文件：
   - **ipaadmin_password** 变量设置的 IdM 管理员密码。
   - IdM 用户的默认 shell 由 **/bin/sh** 中的 **defaultshell** 设置。
5. 保存该文件。
6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 **secret.yml** 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/config/ensure-config-options-are-set.yml
```

**验证步骤**

您可以通过在 IdM 中启动一个新会话来验证默认用户 shell 是否已更改：

1. 以 IdM 管理员身份登录到 **ipaserver** ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 显示当前的 shell：

```
[admin@server /]$ echo "$SHELL"
/bin/sh
```

登录用户正在使用 **sh** shell。

## 10.4. 使用 ANSIBLE 为 IDM 域配置 NETBIOS 名称

NetBIOS 名称用于 Microsoft Windows 的(SMB)类型的共享和消息。您可以使用 NetBIOS 名称映射驱动器或连接到打印机。

按照以下流程，使用 Ansible playbook 为您的身份管理(IdM)域配置 NetBIOS 名称。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - **ansible-freeipa** 软件包已安装。

**假设**

- 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
- 示例假定 **secret.yml** Ansible vault 存储了您的 **ipaadmin_password**，并且您知道 vault 文件密码。

**流程**

1. 进入您的 ***~/MyPlaybooks/*** 目录：

```
$ cd ~/MyPlaybooks/
```

2. 创建一个 *netbios-domain-name-present.yml* Ansible playbook 文件。
3. 在文件中添加以下内容：

```
---
- name: Playbook to change IdM domain netbios name
  hosts: ipaserver
  become: no
  gather_facts: no
```

```
vars_files:
- /home/user_name/MyPlaybooks/secret.yml

tasks:
  - name: Set IdM domain netbios name
    ipaconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      netbios_name: IPADOM
```

4. 保存该文件。

5. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 **secret.yml** 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory netbios-domain-name-present.yml
```

出现提示时，提供 vault 文件密码。

**其它资源**

- 配置 NetBIOS 名称的指南

## 10.5. 使用 ANSIBLE 确保 IDM 用户和组有 SID

身份管理(IdM)服务器可以根据本地域的 ID 范围中的数据，在内部将唯一安全标识符(SID)分配给 IdM 用户和组。SID 存储在用户和组对象中。

确保 IdM 用户和组有 SID 的目标是允许生成特权属性证书(PAC)，这是 IdM-IdM 信任的第一步。如果 IdM 用户和组有 SID，IdM 能够使用 PAC 数据发布 Kerberos 票据。

按照以下流程实现以下目标：

- 为已存在的 IdM 用户和用户组生成 SID。
- 为 IdM 新用户和组启用 SID 生成。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - **ansible-freeipa** 软件包已安装。

**假设**

- 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
- 示例假定 **secret.yml** Ansible vault 存储了您的 **ipaadmin_password**，并且您知道 vault 文件密码。

**流程**

1. 进入您的 ***~/MyPlaybooks/*** 目录：

```
$ cd ~/MyPlaybooks/
```

2. 创建一个 *sids-for-users-and-groups-present.yml* Ansible playbook 文件。

3. 在文件中添加以下内容：

```
---
- name: Playbook to ensure SIDs are enabled and users and groups have SIDs
  hosts: ipaserver
  become: no
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
    - name: Enable SID and generate users and groups SIDS
      ipaconfig:
        ipaadmin_password: "{{ ipaadmin_password }}"
        enable_sid: true
        add_sids: true
```

**enable_sid** 变量为将来的 IdM 用户和组启用 SID 生成。 **add_sids** 变量为现有的 IdM 用户和组生成 SID。

**注意**

使用 **add_sids: true** 时，您还必须将 **enable_sid** 变量设置为 **true**。

4. 保存该文件。

5. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 **secret.yml** 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory sids-for-users-and-groups-present.yml
```

出现提示时，提供 vault 文件密码。

**其它资源**

- IdM ID 范围中的安全性和相对标识符的角色。

## 10.6. 其它资源

- 请参阅 **/usr/share/doc/ansible-freeipa/** 目录中的 **README-config.md**。
- 请参阅 **/usr/share/doc/ansible-freeipa/playbooks/config** 目录中的 playbook 示例。

# 第 11 章 使用命令行管理用户帐户

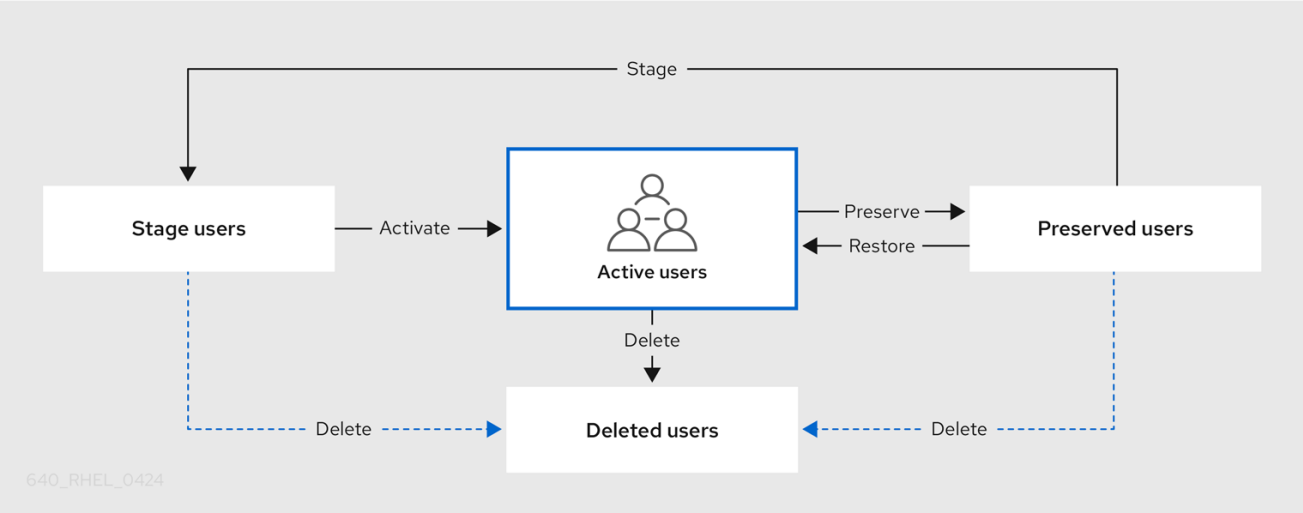
IdM （身份管理）中的用户生命周期有几个阶段，包括：

- 创建用户帐户
- 激活 stage 用户帐户
- 保留用户帐户
- 删除 active、stage 或 preserved 用户帐户
- 恢复 preserved 用户帐户

## 11.1. 用户生命周期

身份管理(IdM)支持三个用户帐户状态：

- **Stage（预发布）** 用户不允许进行身份验证。这是初始状态。活动用户所需的一些用户帐户属性无法在这里设置，例如组成员资格。
- **Active（活跃）** 用户被允许进行身份验证。所有必需的用户帐户属性都需要在这个阶段设置。
- **Preserved（保留）** 用户是以前活跃的用户，但现在被视为不活跃且无法通过 IdM 进行身份验证。保留用户保留他们作为活跃用户的大多数帐户属性，但它们不属于任何用户组。




您可以从 IdM 数据库永久删除用户条目。

**重要**

删除的用户帐户无法恢复。当您删除用户帐户时，与帐户相关的所有信息都将永久丢失。

只能由具备管理员权限的用户（如默认的 admin 用户）才能创建新的管理员。如果您意外删除所有管理员帐户，目录管理器必须在 Directory 服务器中手动创建新管理员。

**警告**

不要删除 **admin** 用户。由于 **admin** 是 IdM 所需的预定义用户，因此此操作会导致某些命令出现问题。如果要定义和使用另外的 admin 用户，请先至少为一个其他用户授予 **admin** 权限，然后再使用 **ipa user-disable admin** 命令来禁用预定义的 admin 用户。

**警告**

不要将本地用户添加到 IdM。NSS（Name Service Switch）在解析本地用户和组前，总会先解析 IdM 的用户和组。这意味着 IdM 组成员资格不适用于本地用户。

## 11.2. 使用命令行添加用户

您可以将用户添加为：

- **Active** - 可以被他们的用户主动使用的用户账户。
- **stage** - 无法使用这些帐户。如果要准备新用户帐户，请使用它。当用户准备好使用其帐户时，您可以激活他们。

以下流程描述了使用 **ipa user-add** 命令将活跃用户添加到 IdM 服务器中。

同样，您可以使用 **ipa stageuser-add** 命令创建stage 用户帐户。

**注意**

IdM 自动给新用户帐户分配唯一的用户 ID(UID)。您也可以手动执行此操作，但服务器不会验证 UID 号是否是唯一的。因此，多个用户条目可能被分配了相同的 ID 号。红帽建议防止多个条目具有相同的 UID 。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 获得 Kerberos ticket。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. 打开终端并连接到 IdM 服务器。
2. 添加用户登录、用户名、姓氏以及可选，您也可以添加其电子邮件地址。

```
$ ipa user-add user_login --first=first_name --last=last_name --email=email_address
```

IdM 支持可通过以下正则表达式描述的用户名：

```
[a-zA-Z0-9_.][a-zA-Z0-9_.-]{0,252}[a-zA-Z0-9_.$-]?
```

**注意**

支持以末尾的美元符号($)结尾的用户名，以启用 Samba 3.x 机器支持。

如果您添加了包含大写字符的用户名，IdM 会在保存名称时自动将其转换为小写。因此，IdM 总是需要在登录时以小写形式输入用户名。此外，不能添加仅在字母大小写上不同的用户名，比如 **user** 和 **User**。

用户名的默认最大长度为 32 个字符。要更改它，请使用 **ipa config-mod --maxusername** 命令。例如，要将最大用户名长度增加到 64 个字符：

```
$ ipa config-mod --maxusername=64
 Maximum username length: 64
 ...
```

**ipa user-add** 命令包含许多参数。要全部列出它们，请使用 ipa help 命令：

```
$ ipa help user-add
```

有关 **ipa help** 命令的详情，请查看 什么是 IPA help 。

您可以通过列出所有 IdM 用户帐户来验证新用户帐户是否已成功创建：

```
$ ipa user-find
```

此命令列出所有用户帐户及详细信息。

## 11.3. 使用命令行激活用户

要通过将用户帐户从 stage 移到 active 来激活它，， 请使用 **ipa stageuser-activate** 命令。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 获得 Kerberos ticket。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. 打开终端并连接到 IdM 服务器。
2. 使用以下命令激活用户帐户：

```
$ ipa stageuser-activate user_login
-------------------------
Stage user user_login activated
-------------------------
...
```

您可以通过列出所有 IdM 用户帐户来验证新用户帐户是否已成功创建：

```
$ ipa user-find
```

此命令列出所有用户帐户及详细信息。

## 11.4. 使用命令行保留用户

如果要删除用户帐户，您可以保留该帐户，保留这个选项以便以后恢复。要保留用户帐户，请使用 **ipa user-del** 或 **ipa stageuser-del** 命令的 **--preserve** 选项。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 获得 Kerberos ticket。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. 打开终端并连接到 IdM 服务器。
2. 使用以下命令保留用户帐户：

```
$ ipa user-del --preserve user_login
--------------------
Deleted user "user_login"
--------------------
```

**注意**

尽管输出说用户帐户已删除，但实际上是被保留了。

## 11.5. 使用命令行删除用户

IdM（身份管理）可让您永久删除用户。您可以删除：

- 活动用户,使用以下命令：**ipa user-del**
- Stage 用户,使用以下命令：**ipa stageuser-del**
- Preserved 用户，使用以下命令：**ipa user-del**

删除多个用户时，请使用 **--continue** 选项强制命令继续，而不论出现什么错误。命令完成后，会将成功和失败的操作摘要输出到 **stdout** 标准输出流。

```
$ ipa user-del --continue user1 user2 user3
```

如果不使用 **--continue**，命令会继续删除用户，直到它遇到错误，然后它会停止并退出。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 获得 Kerberos ticket。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. 打开终端并连接到 IdM 服务器。
2. 使用以下命令删除用户帐户：

```
$ ipa user-del user_login
--------------------
Deleted user "user_login"
--------------------
```

用户帐户从 IdM 永久删除。

## 11.6. 使用命令行恢复用户

您可以将 preserved 用户恢复成：

- Active 用户：**ipa user-undel**
- Stage 用户：**ipa user-stage**

恢复用户帐户不会恢复帐户之前的所有属性。例如，用户的密码不会被恢复，必须再次设置。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 获得 Kerberos ticket。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. 打开终端并连接到 IdM 服务器。
2. 使用以下命令激活用户帐户：

```
$ ipa user-undel user_login
----------------------------
Undeleted user account "user_login"
----------------------------
```

或者，您可以将用户帐户恢复为暂存的用户帐户：

```
$ ipa user-stage user_login
----------------------------
Staged user account "user_login"
----------------------------
```

**验证步骤**

- 您可以通过列出所有 IdM 用户帐户来验证新用户帐户是否已成功创建：

```
$ ipa user-find
```

此命令列出所有用户帐户及详细信息。

# 第 12 章 使用 IDM WEB UI 管理用户帐户

身份管理(IdM)提供 多个阶段，可帮助您管理各种用户生命周期情况：

**创建用户帐户**

在员工在公司开始职业生涯之前 创建 stage 用户帐户，并提前在员工出现在办公室并想要激活客户的那天前做好准备。
您可以省略此步骤，并直接创建活动的用户帐户。这个流程与创建 stage 用户帐户的流程类似。

**激活用户帐户**

激活帐户 在员工的第一个工作日。

**禁用用户帐户**

如果用户要休几个月的产假，您需要 临时禁用该帐户。

**启用用户帐户**

用户返回时，您需要 重新启用该帐户。

**保留用户帐户**

如果用户想要离开公司，您需要删除该 帐户，并有可能恢复它， 因为人们可以在一段时间后回到公司。

**恢复用户帐户**

两年后，用户回来了，您需要 恢复保留的帐户。

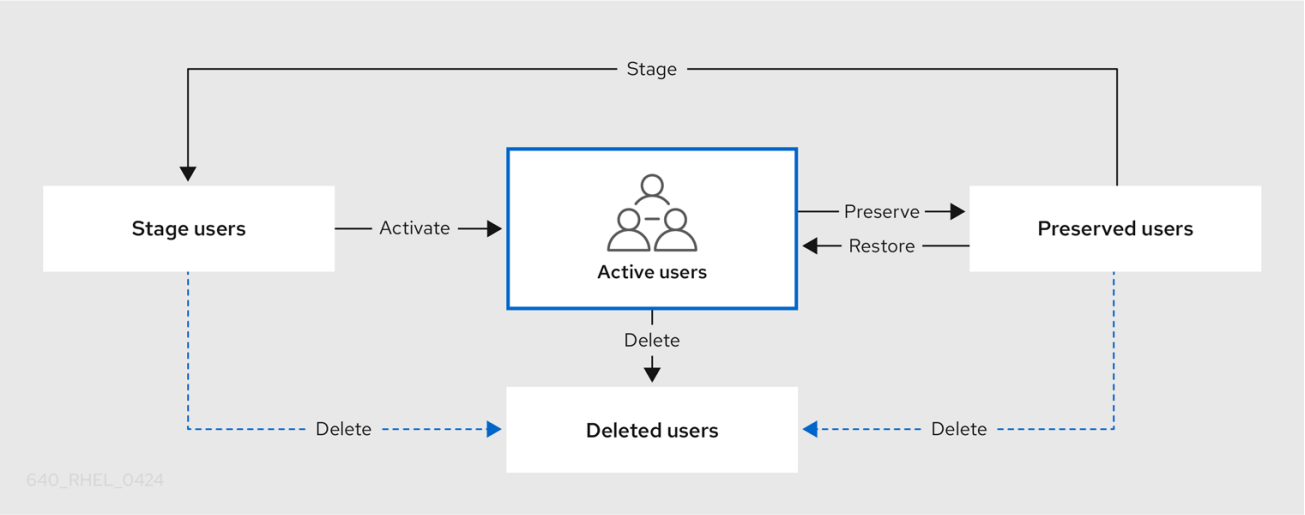
**删除用户帐户**

如果取消了员工，请在没有备份的情况下删除帐户。

## 12.1. 用户生命周期

身份管理(IdM)支持三个用户帐户状态：

- **Stage（预发布）** 用户不允许进行身份验证。这是初始状态。活动用户所需的一些用户帐户属性无法在这里设置，例如组成员资格。
- **Active（活跃）** 用户被允许进行身份验证。所有必需的用户帐户属性都需要在这个阶段设置。
- **Preserved（保留）** 用户是以前活跃的用户，但现在被视为不活跃且无法通过 IdM 进行身份验证。保留用户保留他们作为活跃用户的大多数帐户属性，但它们不属于任何用户组。




您可以从 IdM 数据库永久删除用户条目。

**重要**

删除的用户帐户无法恢复。当您删除用户帐户时，与帐户相关的所有信息都将永久丢失。

只能由具备管理员权限的用户（如默认的 admin 用户）才能创建新的管理员。如果您意外删除所有管理员帐户，目录管理器必须在 Directory 服务器中手动创建新管理员。

**警告**

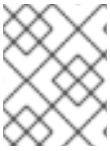
不要删除 **admin** 用户。由于 **admin** 是 IdM 所需的预定义用户，因此此操作会导致某些命令出现问题。如果要定义和使用另外的 admin 用户，请先至少为一个其他用户授予 **admin** 权限，然后再使用 **ipa user-disable admin** 命令来禁用预定义的 admin 用户。

**警告**

不要将本地用户添加到 IdM。NSS（Name Service Switch）在解析本地用户和组前，总会先解析 IdM 的用户和组。这意味着 IdM 组成员资格不适用于本地用户。

## 12.2. 在 WEB UI 中添加用户

通常，您需要在新员工开始工作前创建新的用户帐户。这样的 stage 帐户无法访问，您需要之后将其激活。



**注意**

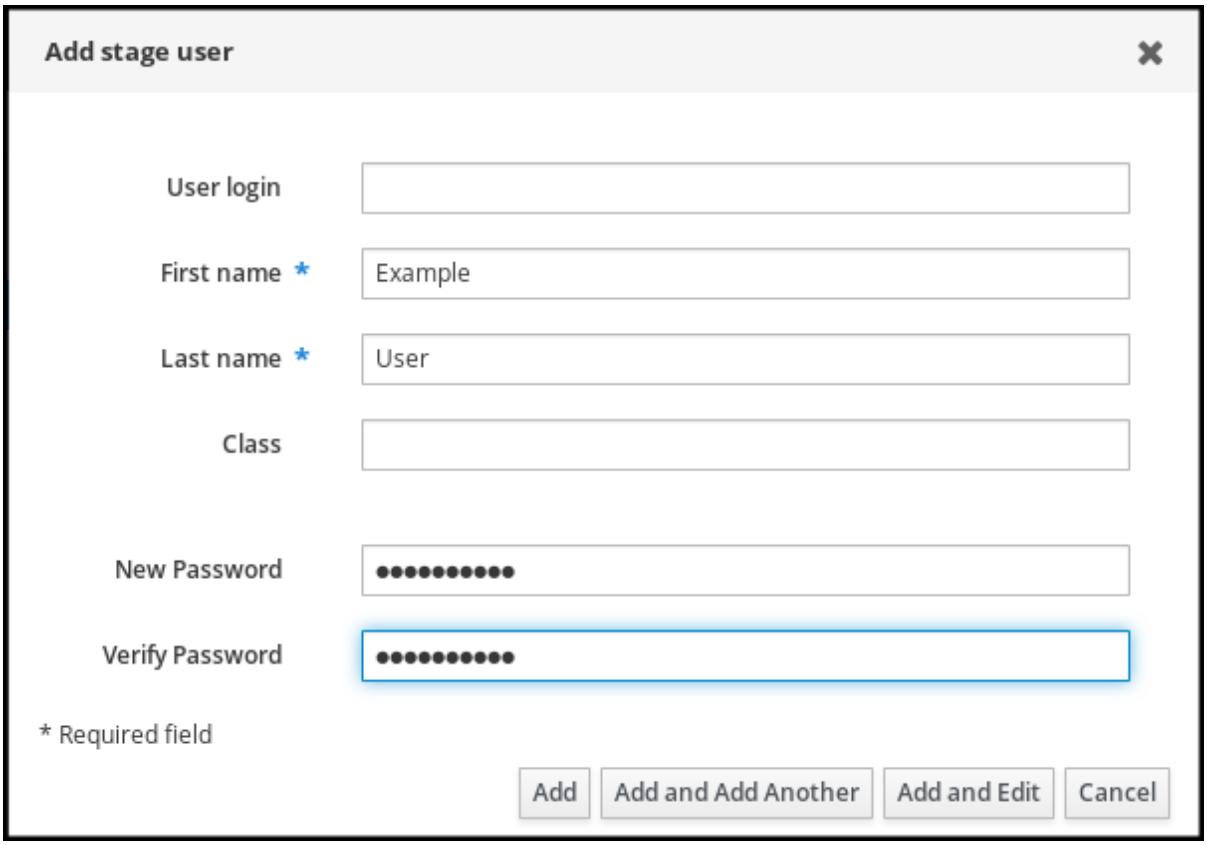
或者，您可以直接创建活动的用户帐户。要添加活动的用户，请按照下面的流程，并在 **Active users** 选项卡中添加用户帐户。

**先决条件**

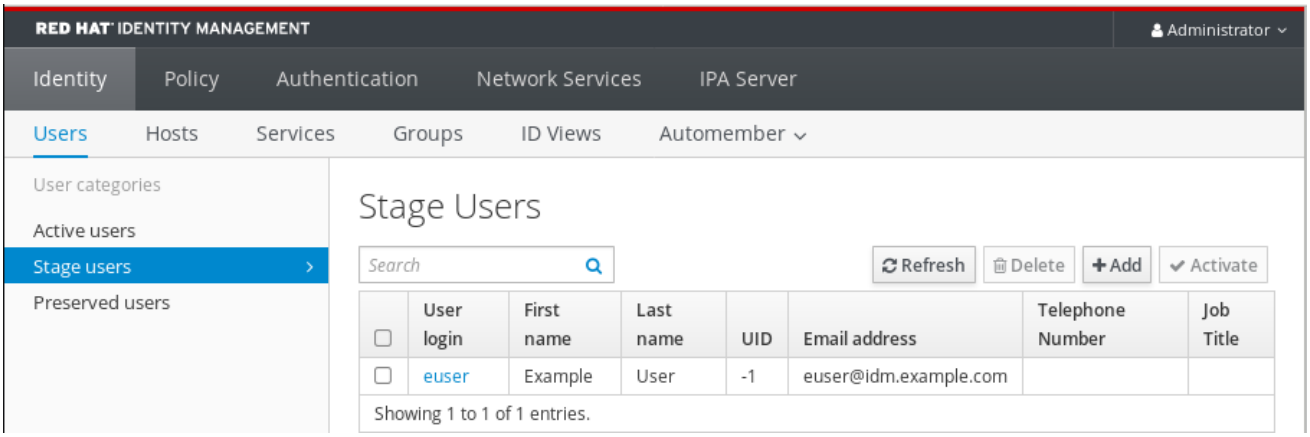
- 管理 IdM 或用户管理员角色的管理员特权。

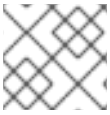
**流程**

1. 登录到 IdM Web UI。
   详情请参阅 在 Web 浏览器中访问 IdM Web UI 。
2. 进到 **Users → Stage Users** 选项卡。
   另外，您可以在 **Users → Active users** 中添加用户帐户，但是您无法将用户组添加到帐户中。
3. 单击 **+ Add** 图标。
4. 在 **Add stage user** 对话框中，输入新用户的 **First name** 和 **Last name**。
5. [可选] 在 **User login** 字段中，添加一个登录名称。
   如果您将其留空，IdM 服务器将以以下形式创建登录名称：名字的第一个字母和姓氏。整个登录名最多可有 32 个字符。
6. [可选] 在 GID 下拉菜单中，选择应包含该用户的组。
7. [可选] 在 **Password** 和 **Verify password** 字段中，输入您的密码并确认，确保它们都匹配。
8. 点击 **Add** 按钮。




此时，您可以在 **Stage Users** 表中看到用户帐户。



**注意**

如果点击用户名，您可以编辑高级设置，如添加电话号码、地址或职业。

## 12.3. 在 IDM WEB UI 中 STAGE 用户

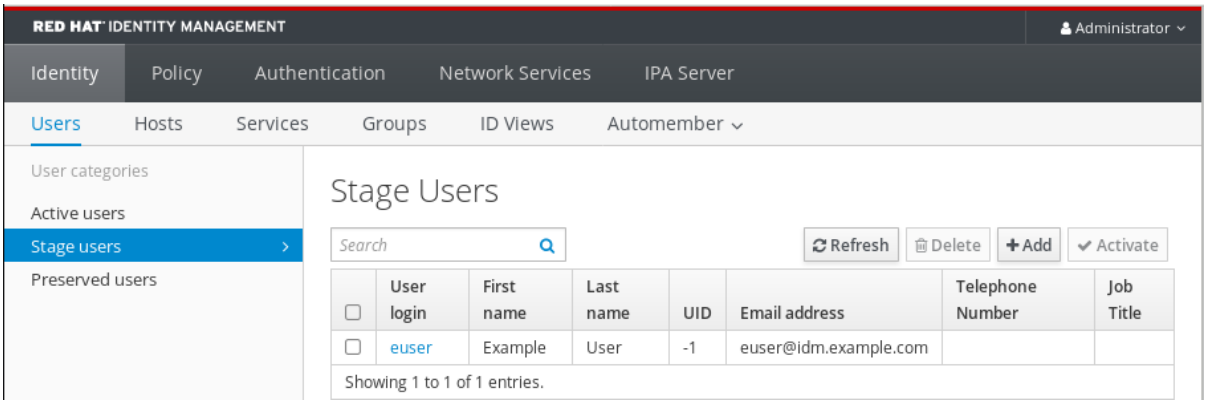
在用户可以登录到 IdM 之前以及将用户添加到 IdM 组之前，您必须按照这个流程激活 stage 用户帐户。

**先决条件**

- 管理 IdM Web UI 或用户管理员角色的管理员特权。
- IdM 中至少有一个 stage 用户帐户。

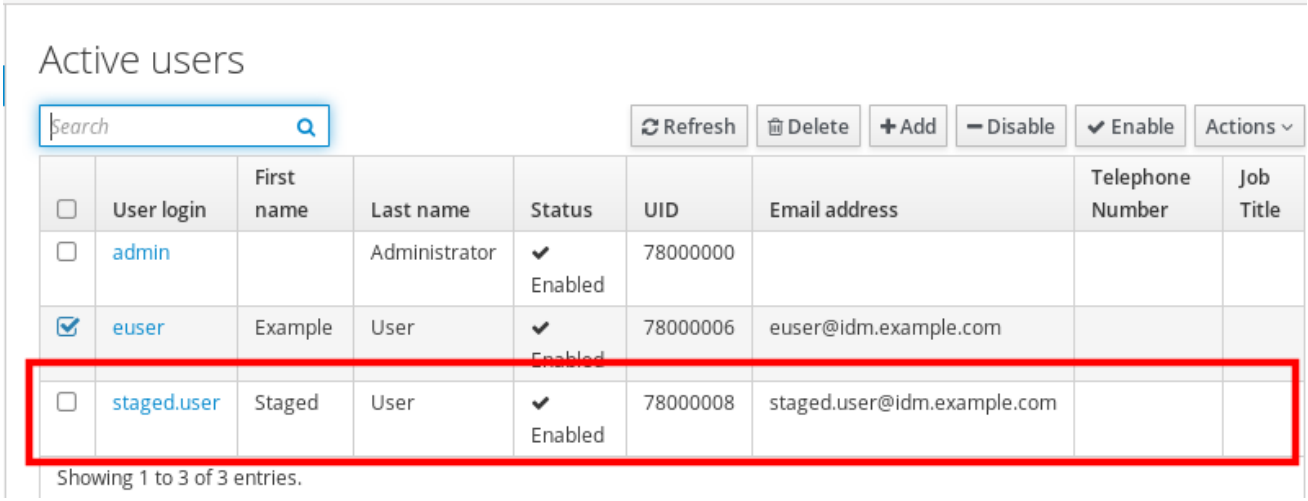
**流程**

1. 登录到 IdM Web UI。
   详情请参阅 在 Web 浏览器中访问 IdM Web UI 。
2. 进到 **Users → Stage users** 选项卡。
3. 单击您要激活的用户帐户的复选框。
4. 单击 **Activate** 按钮。




5. 在 **Confirmation** 对话框中，单击 **OK**。

如果激活成功，IdM Web UI 会显示绿色的确认信息，表示用户已激活，并且用户帐户已移到 **Active 用户**。帐户处于活动状态，用户才可以向 IdM 域和 IdM Web UI 进行身份验证。在第一次登录时，系统将提示用户更改密码。




**注意**

在此阶段，您可以向用户组添加活动的用户帐户。

## 12.4. 在 WEB UI 中禁用用户帐户

您可以禁用活动的用户帐户。禁用用户帐户会停用该帐户，因此用户帐户无法进行身份验证，并使用 IdM 服务，如 Kerberos 或执行任何任务。

禁用的用户帐户仍然在 IdM 中存在，所有相关信息保持不变。与保留的用户帐户不同，禁用的用户帐户保持活动状态，并且可以是用户组的成员。

**注意**

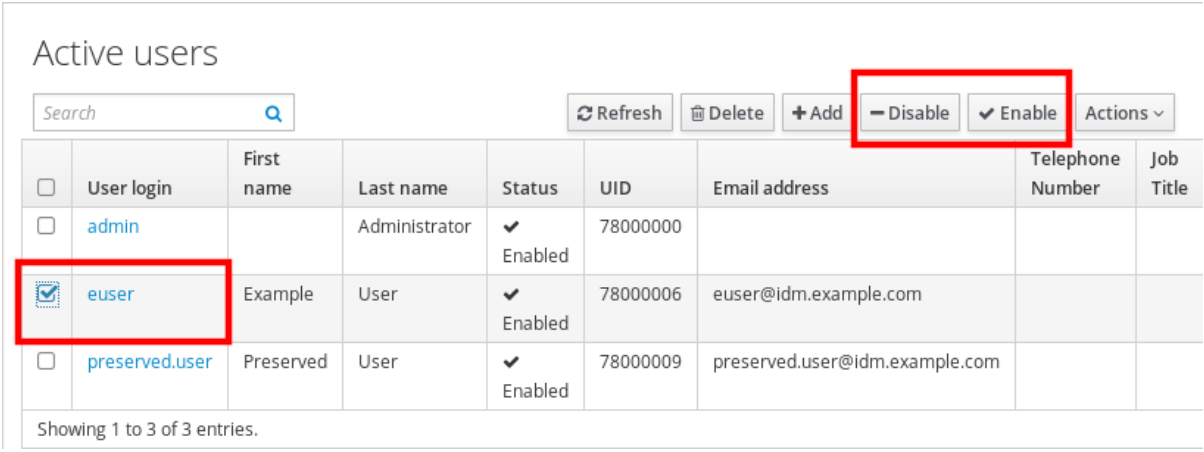
禁用用户帐户后，任何现有的连接都会保持有效，直到用户的 Kerberos TGT 和其他票据过期为止。票据过期后，用户将无法续订。

**先决条件**

- 管理 IdM Web UI 或用户管理员角色的管理员特权。

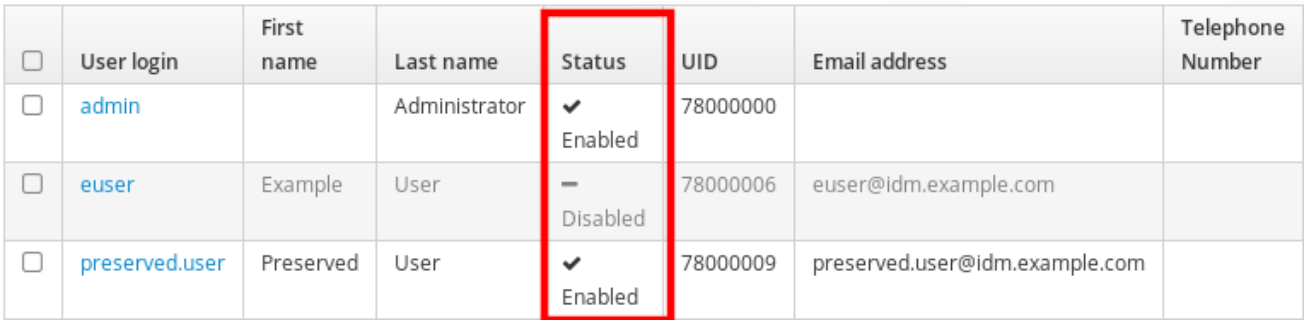
**流程**

1. 登录到 IdM Web UI。
   详情请参阅 在 Web 浏览器中访问 IdM Web UI 。
2. 进到 **Users → Active users** 选项卡。
3. 点击您要禁用的用户帐户的复选框。
4. 单击 **Disable** 按钮。




5. 在 **Confirmation** 对话框中，单击 **OK** 按钮。

如果禁用过程成功，您可以在 **Active users** 表中的 Status 列中验证。




## 12.5. 在 WEB UI 中启用用户帐户

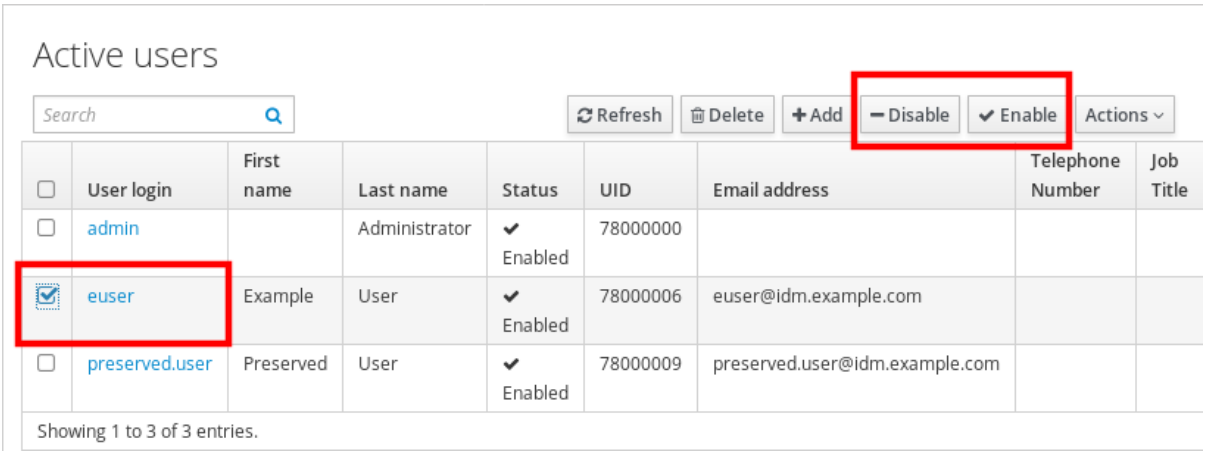
通过 IdM，您可以启用禁用的活动用户帐户。启用用户帐户可激活禁用的帐户。

**先决条件**

- 管理 IdM Web UI 或用户管理员角色的管理员特权。

**流程**

1. 登录到 IdM Web UI。
2. 进到 **Users → Active users** 选项卡。
3. 单击您要启用的用户帐户的复选框。
4. 单击 **Enable** 按钮。




5. 在 **Confirmation** 对话框中，单击 **OK** 按钮。

如果更改成功，您可以在 **Active users** 表中的 Status 列中验证。

## 12.6. 在 IDM WEB UI 中保留活动的用户

保留用户帐户可让您从 **Active users** 选项卡中删除帐户，而将这些帐户保留在 IdM 中。

如果员工离开了公司，可保留用户帐户。如果您要禁用用户帐户数周或数月（例如，产假），请禁用该帐户。详情请参阅 在 Web UI 中禁用用户帐户。保留的帐户不是活动的，用户无法使用它们访问内部网络，但该帐户及所有数据都保留在数据库中。

您可以将恢复的帐户移回到活动模式。

**注意**

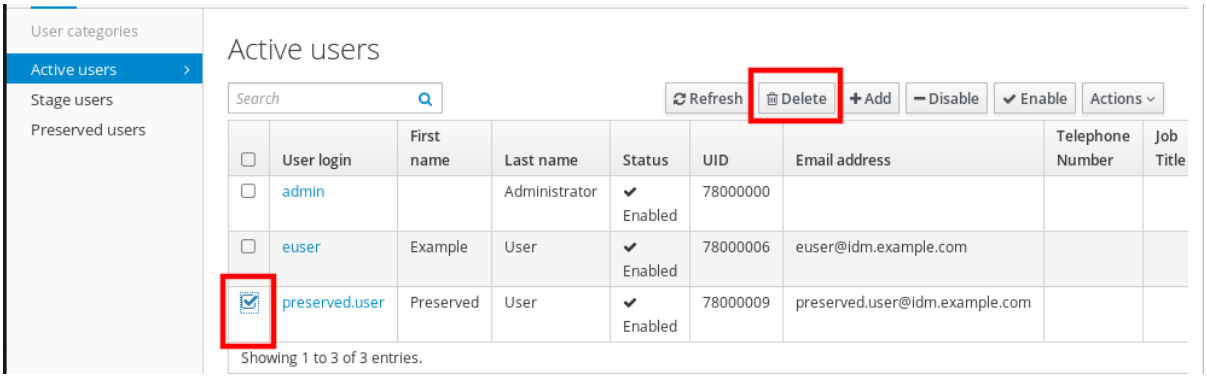
处于保留状态的用户列表可以提供过去用户帐户的历史记录。

**先决条件**

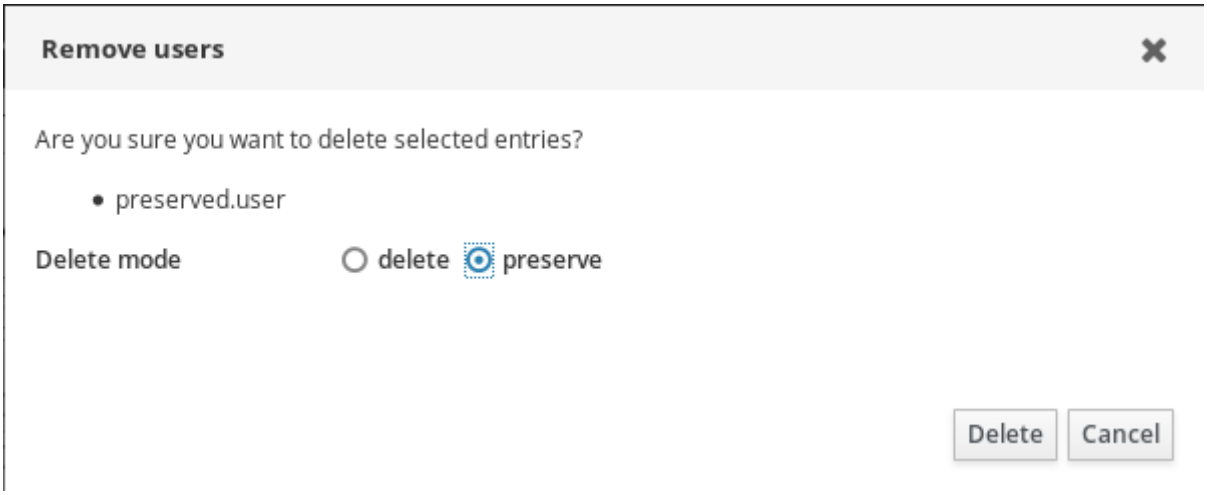
- 管理 IdM（身份管理）Web UI 或用户管理员角色的管理员特权。

**流程**

1. 登录到 IdM Web UI。
   详情请参阅 在 Web 浏览器中访问 IdM Web UI。
2. 进到 **Users → Active users** 选项卡。
3. 单击您要保留的用户帐户的复选框。
4. 单击 **Delete** 按钮。




5. 在 **Remove users** 对话框中，将 **Delete mode** 单选按钮切换到 **preserve**。
6. 单击 **Delete** 按钮。




因此，用户帐户被移到 **Preserved users**。

如果需要恢复保留的用户，请参阅 在 IdM Web UI 中恢复用户 。

## 12.7. 在 IDM WEB UI 中恢复用户

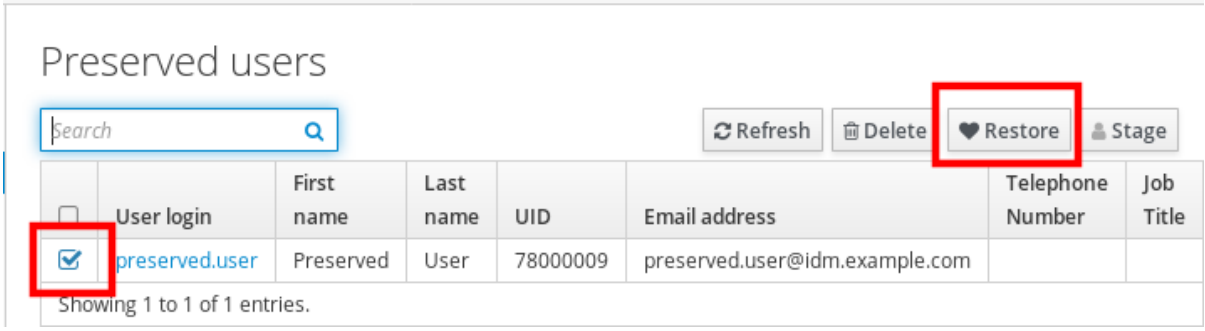
IdM （身份管理）可让您将保留的用户帐户恢复到 active 状态。您可以将 preserved 用户恢复到活跃用户或 stage 用户。

**先决条件**

- 管理 IdM Web UI 或用户管理员角色的管理员特权。

**流程**

1. 登录到 IdM Web UI。
   详情请参阅 在 Web 浏览器中访问 IdM Web UI 。
2. 进到 **Users → Preserved users** 选项卡。
3. 单击您要恢复的用户帐户的复选框。
4. 单击 **Restore** 按钮。




5. 在 **Confirmation** 对话框中，单击 **OK** 按钮。

IdM Web UI 显示一条绿色确认信息，并将用户帐户移到 **Active users** 选项卡中。

## 12.8. 在 IDM WEB UI 中删除用户

删除用户是一种不可逆的操作，导致用户帐户被从 IdM 数据库中永久删除，包括组成员资格和密码。任何对用户的外部配置，如系统帐户和主目录，都不会被删除，但无法通过 IdM 来访问。

您可以删除：

- Active 用户 - IdM Web UI 为您提供了选项：
  - 临时保留用户
    详情请查看 在 IdM Web UI 中保留活动用户 。
  - 永久删除它们
- Stage 用户 - 您可以永久删除 stage 用户。
- Preserved 用户 - 您可以永久删除 preserved 用户。

以下流程描述了删除活动用户。同样，您可以删除用户帐户，在：

- **Stage users** 选项卡
- **Preserved users** 选项卡

**先决条件**

- 管理 IdM Web UI 或用户管理员角色的管理员特权。

**流程**

1. 登录到 IdM Web UI。
   详情请参阅 在 Web 浏览器中访问 IdM Web UI 。
2. 进到 **Users → Active users** 选项卡。
   或者，您可以在 **Users → Stage users** 或 **Users → Preserved users** 删除用户账户。
3. 点 **Delete** 图标。
4. 在 **Remove users** 对话框中，将 **Delete mode** 单选按钮切换到 **delete**。
5. 单击 **Delete** 按钮。

用户帐户从 IdM 永久删除。

# 第 13 章 使用 ANSIBLE PLAYBOOK 管理用户帐户

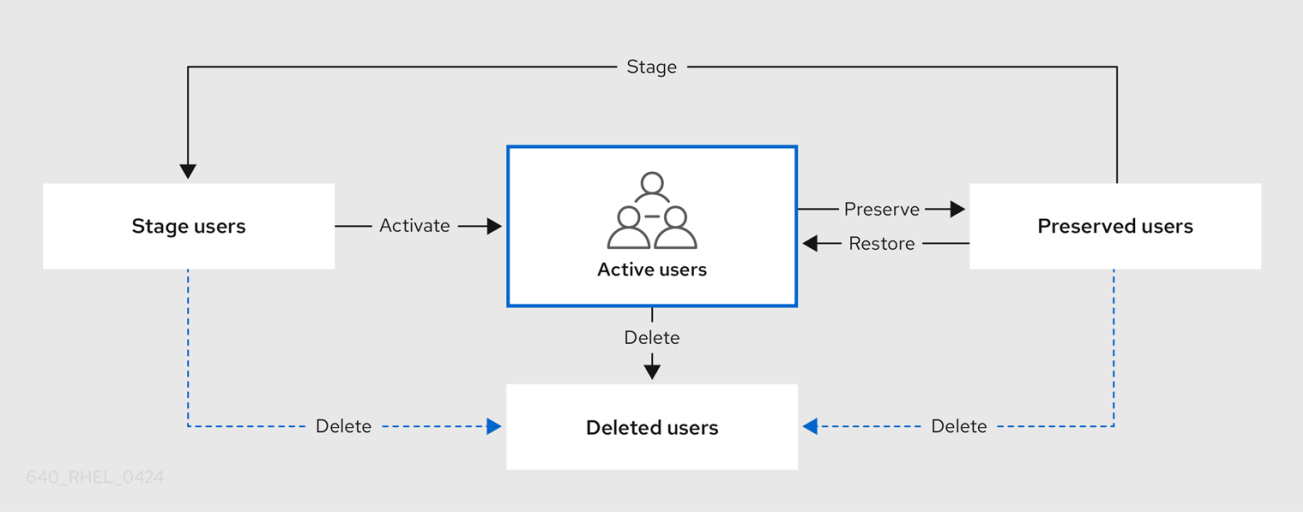
您可以使用 Ansible playbook 管理 IdM 中的用户。在介绍了用户生命周期后，本章将介绍如何将 Ansible playbook 用于以下操作：

- 确保直接在 **YML** 文件中列出的单个用户存在。
- 确保直接在 **YML** 文件中列出的多个用户存在。
- 确保从 **YML** 文件引用的 **JSON** 文件中列出的多个用户存在。
- 确保直接在 **YML** 文件中列出的用户不存在。

## 13.1. 用户生命周期

身份管理(IdM)支持三个用户帐户状态：

- **Stage（预发布）** 用户不允许进行身份验证。这是初始状态。活动用户所需的一些用户帐户属性无法在这里设置，例如组成员资格。
- **Active（活跃）** 用户被允许进行身份验证。所有必需的用户帐户属性都需要在这个阶段设置。
- **Preserved（保留）** 用户是以前活跃的用户，但现在被视为不活跃且无法通过 IdM 进行身份验证。保留用户保留他们作为活跃用户的大多数帐户属性，但它们不属于任何用户组。




您可以从 IdM 数据库永久删除用户条目。

**重要**

删除的用户帐户无法恢复。当您删除用户帐户时，与帐户相关的所有信息都将永久丢失。

只能由具备管理员权限的用户（如默认的 admin 用户）才能创建新的管理员。如果您意外删除所有管理员帐户，目录管理器必须在 Directory 服务器中手动创建新管理员。

**警告**

不要删除 **admin** 用户。由于 **admin** 是 IdM 所需的预定义用户，因此此操作会导致某些命令出现问题。如果要定义和使用另外的 admin 用户，请先至少为一个其他用户授予 **admin** 权限，然后再使用 **ipa user-disable admin** 命令来禁用预定义的 admin 用户。

**警告**

不要将本地用户添加到 IdM。NSS（Name Service Switch）在解析本地用户和组前，总会先解析 IdM 的用户和组。这意味着 IdM 组成员资格不适用于本地用户。

## 13.2. 使用 ANSIBLE PLAYBOOK 确保存在一个 IDM 用户

以下流程描述了确保使用 Ansible playbook 在 IdM 中存在用户。

**先决条件**

- 您需要知道 IdM **admin** 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 *~/MyPlaybooks/* 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible 库存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 创建一个清单文件，如 **inventory.file**，并在该文件中定义 **ipaserver** ：

```
[ipaserver]
server.idm.example.com
```

2. 创建一个 Ansible playbook 文件，其中包含您要确保的 IdM 中存在的用户数据。要简化此步骤，您可以复制并修改 **/usr/share/doc/ansible-freeipa/playbooks/user/add-user.yml** 文件中的示例。例如，创建名为 *idm_user* 的用户并添加 *Password123* 作为用户密码：

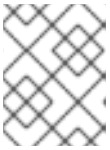
```
---
```

```
- name: Playbook to handle users
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Create user idm_user
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: idm_user
      first: Alice
      last: Acme
      uid: 1000111
      gid: 10011
      phone: "+555123457"
      email: idm_user@acme.com
      passwordexpiration: "2023-01-19 23:59:59"
      password: "Password123"
      update_password: on_create
```

您必须使用以下选项来添加用户：

- **name**：登录名称
- **first**：名（字符串）
- **last**：姓（字符串）

有关可用用户选项的完整列表，请参阅 **/usr/share/doc/ansible-freeipa/README-user.md** Markdown 文件。



**注意**

如果您使用 **update_password: on_create** 选项，Ansible 仅在创建用户时创建用户密码。如果已使用密码创建了用户，Ansible 不会生成新的密码。

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-IdM-
user.yml
```

**验证步骤**

- 您可以使用 **ipa user-show** 命令验证 IdM 中是否存在新用户帐户：
  1. 以 admin 用户身份登录 **ipaserver** ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

  2. 为 admin 请求一个 Kerberos ticket：

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. 请求有关 *idm_user* 的信息：

```
$ ipa user-show idm_user
  User login: idm_user
  First name: Alice
  Last name: Acme
  ....
```

IdM 中存在名为 *idm_user* 的用户。

## 13.3. 使用 ANSIBLE PLAYBOOK 确保存在多个 IDM 用户

以下流程描述了使用 Ansible playbook 确定在 IdM 中存在多个用户。

**先决条件**

- 您需要知道 IdM **admin** 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible 库存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 创建一个清单文件，如 **inventory.file**，并在该文件中定义 **ipaserver** ：

```
[ipaserver]
server.idm.example.com
```

2. 创建一个 Ansible playbook 文件，其中包含您要在 IdM 中确保存在的用户的数据。要简化此步骤，您可以复制并修改 **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml** 文件中的示例。例如，要创建用户 *idm_user_1*、*idm_user_2* 和 *idm_user_3*，并添加 *Password123* 作为密码 *idm_user_1* ：

```
---
- name: Playbook to handle users
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
```

```
- name: Create user idm_users
  ipauser:
    ipaadmin_password: "{{ ipaadmin_password }}"
    users:
    - name: idm_user_1
      first: Alice
      last: Acme
      uid: 10001
      gid: 10011
      phone: "+555123457"
      email: idm_user@acme.com
      passwordexpiration: "2023-01-19 23:59:59"
      password: "Password123"
    - name: idm_user_2
      first: Bob
      last: Acme
      uid: 100011
      gid: 10011
    - name: idm_user_3
      first: Eve
      last: Acme
      uid: 1000111
      gid: 10011
```

**注意**

如果没有指定 **update_password: on_create** 选项，Ansible 每次运行 playbook 时都会重新设置用户密码：如果用户自上次运行 playbook 起更改了密码，则 Ansible 重新设置密码。

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-
users.yml
```

**验证步骤**

- 您可以使用 **ipa user-show** 命令验证用户帐户是否存在于 IdM 中：

1. 以管理员身份登录到 **ipaserver**：

```
$ ssh administrator@server.idm.example.com
Password:
[admin@server /]$
```

2. 显示有关 *idm_user_1* 的信息：

```
$ ipa user-show idm_user_1
  User login: idm_user_1
  First name: Alice
  Last name: Acme
  Password: True
  ....
```

IdM 中存在名为 *idm_user_1* 的用户。

## 13.4. 使用 ANSIBLE PLAYBOOK 确保存在 JSON 文件中的多个 IDM 用户

以下流程描述了如何使用 Ansible playbook 确保在 IdM 中存在多个用户。用户存储在 **JSON** 文件中。

**先决条件**

- 您需要知道 IdM **admin** 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible 库存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 创建一个清单文件，如 **inventory.file**，并在该文件中定义 **ipaserver** ：

```
[ipaserver]
server.idm.example.com
```

2. 创建包含必要任务的 Ansible playbook 文件。使用您要确保存在的用户数据引用 **JSON** 文件。要简化此步骤，您可以复制并修改 **/usr/share/doc/ansible-freeipa/ensure-users-present-ymlfile.yml** 文件中的示例：

```
---
- name: Ensure users' presence
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Include users.json
    include_vars:
      file: users.json

  - name: Users present
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      users: "{{ users }}"
```

3. 创建 **users.json** 文件，并将 IdM 用户添加到其中。要简化此步骤，您可以复制并修改 **/usr/share/doc/ansible-freeipa/playbooks/user/users.json** 文件中的示例。例如，要创建用户 *idm_user_1*、*idm_user_2* 和 *idm_user_3*，并添加 *Password123* 作为密码 *idm_user_1* ：

```
{
  "users": [
   {
    "name": "idm_user_1",
    "first": "Alice",
    "last": "Acme",
    "password": "Password123"
   },
   {
    "name": "idm_user_2",
    "first": "Bob",
    "last": "Acme"
   },
   {
    "name": "idm_user_3",
    "first": "Eve",
    "last": "Acme"
   }
  ]
}
```

4. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 **secret.yml** 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-users-
present-jsonfile.yml
```

**验证步骤**

- 您可以使用 **ipa user-show** 命令验证 IdM 中是否存在用户帐户：

1. 以管理员身份登录到 **ipaserver**：

```
$ ssh administrator@server.idm.example.com
Password:
[admin@server /]$
```

2. 显示有关 *idm_user_1* 的信息：

```
$ ipa user-show idm_user_1
  User login: idm_user_1
  First name: Alice
  Last name: Acme
  Password: True
  ....
```

IdM 中存在名为 *idm_user_1* 的用户。

## 13.5. 确保没有用户使用 ANSIBLE PLAYBOOK

以下流程描述了如何使用 Ansible playbook 来确保 IdM 中没有特定用户。

**先决条件**

- 您需要知道 IdM **admin** 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible 库存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 创建一个清单文件，如 **inventory.file**，并在该文件中定义 **ipaserver** ：

```
[ipaserver]
server.idm.example.com
```

2. 创建 Ansible playbook 文件，使其包含没有 IdM 的用户。要简化此步骤，您可以复制并修改 **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml** 文件中的示例。例如，要删除用户 *idm_user_1*、*idm_user_2* 和 *idm_user_3* ：

```
---
- name: Playbook to handle users
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Delete users idm_user_1, idm_user_2, idm_user_3
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      users:
      - name: idm_user_1
      - name: idm_user_2
      - name: idm_user_3
      state: absent
```

3. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 **secret.yml** 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/delete-
users.yml
```

**验证步骤**

您可以使用 **ipa user-show** 命令验证 IdM 中是否不存在用户帐户：

1. 以管理员身份登录到 **ipaserver**：

```
$ ssh administrator@server.idm.example.com
Password:
[admin@server /]$
```

2. 请求有关 *idm_user_1* 的信息：

```
$ ipa user-show idm_user_1
ipa: ERROR: idm_user_1: user not found
```

IdM 中不存在名为 *idm_user_1* 的用户。

## 13.6. 其它资源

- 请参阅 **/usr/share/doc/ansible-freeipa/** 目录中的 **README-user.md** Markdown 文件。
- 请参阅 **/usr/share/doc/ansible-freeipa/playbooks/user** 目录中的 Ansible playbook 示例。

# 第 14 章 在 IDM CLI 中管理用户组

本章介绍了使用 IdM CLI 的用户组管理。

用户组是一组具有常见特权、密码策略和其他特征的用户。

Identity Management (IdM) 中的用户组可以包括：

- IdM 用户
- 其他 IdM 用户组
- 外部用户，即 IdM 之外的用户

## 14.1. IDM 中的不同组类型

IdM 支持以下类型的组：

**POSIX 组（默认）**

POSIX 组支持其成员的 Linux POSIX 属性。请注意，与 Active Directory 交互的组无法使用 POSIX 属性。
POSIX 属性将用户识别为单独的实体。与用户相关的 POSIX 属性示例包括 **uidNumber**（一个用户号 (UID)）和 **gidNumber**（一个组号 (GID)）。

**非 POSIX 组**

非 POSIX 组不支持 POSIX 属性。例如，这些组没有定义 GID。
这种组的所有成员必须属于 IdM 域。

**外部组**

使用外部组添加存在于 IdM 域外部的身份存储中的组成员，例如：

- 本地系统
- Active Directory 域
- 目录服务

外部组不支持 POSIX 属性。例如，这些组没有定义 GID。

**表 14.1. 默认创建的用户组**

| 组名称 | 默认组成员 |
| --- | --- |
| **ipausers** | 所有 IdM 用户 |
| **admins** | 具有管理特权的用户，包括默认的 **admin** 用户 |
| **editors** | 这是一个旧的组，不再具有任何特殊权限 |
| **trust admins** | 具有管理 Active Directory 信任权限的用户 |

将用户添加到用户组时，该用户将获得与组关联的特权和策略。例如，若要向用户授予管理特权，可将该用户添加到 **admins** 组。

**警告**

不要删除 **admins** 组。由于 **admins** 是 IdM 要求的预定义组，因此此操作会导致某些命令出现问题。

另外，当在 IdM 中创建新用户时，IdM 默认会创建*用户私有组*。有关私有组的更多信息，请参阅在没有私有组的情况下添加用户。

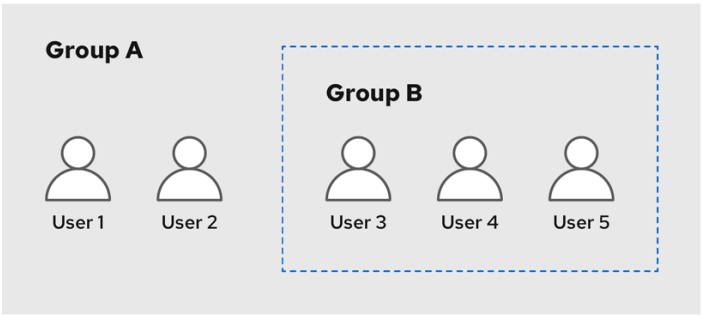
## 14.2. 直接和间接组成员

IdM 中的用户组属性适用于直接和间接成员：当组 B 是组 A 的成员时，组 B 中的所有用户都被视为组 A 的间接成员。

例如，在下图中：

- 用户 1 和用户 2 是组 A 的*直接成员*。
- 用户 3、用户 4 和用户 5 是组 A 的*间接成员*。

**图 14.1. 直接和间接组成员身份**




如果您为用户组 A 设置密码策略，该策略也适用于用户组 B 中的所有用户。

## 14.3. 使用 IDM CLI 添加用户组

按照以下流程，使用 IdM CLI 添加用户组。

**先决条件**

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

- 使用 **ipa group-add *group_name*** 命令添加用户组。例如，创建 group_a：

```
$ ipa group-add group_a
---------------------
Added group "group_a"
---------------------
  Group name: group_a
  GID: 1133400009
```

默认情况下，**ipa group-add** 添加 POSIX 用户组。要指定不同的组类型，请在 **ipa group-add** 中添加选项：

- **--nonposix** 用来创建非 POSIX 组
- **--external** 用来创建外部组
  有关组类型的详情，请查看 IdM 中不同的组类型。

您可以使用 **--gid=*custom_GID*** 选项来在添加用户组时指定自定义的 GID。如果您这样做，请小心以避免 ID 冲突。如果没有指定自定义的 GID，IdM 会自动从可用的 ID 范围内分配一个 GID。

## 14.4. 使用 IDM CLI 搜索用户组

按照以下流程，使用 IdM CLI 搜索现有用户组。

**流程**

- 使用 **ipa group-find** 命令显示所有用户组。要指定组类型，请在 **ipa group-find** 中添加选项：
  - 使用 **ipa group-find --posix** 命令显示所有 POSIX 组。
  - 使用 **ipa group-find --nonposix** 命令显示所有非 POSIX 组。
  - 使用 **ipa group-find --external** 命令显示所有外部组。
    有关不同组类型的更多信息，请参阅 IdM 中的不同组类型。

## 14.5. 使用 IDM CLI 删除用户组

按照以下流程，使用 IdM CLI 删除用户组。请注意，删除组不会从 IdM 中删除组成员。

**先决条件**

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

- 使用 **ipa group-del *group_name*** 命令删除用户组。例如，要删除 group_a：

```
$ ipa group-del group_a
--------------------------
Deleted group "group_a"
--------------------------
```

## 14.6. 使用 IDM CLI 将成员添加到用户组中

您可以将用户和用户组添加为用户组的成员。如需更多信息，请参阅 IdM 中不同的组类型 以及 直接和间接组成员。按照以下流程，使用 IdM CLI 将成员添加到用户组中。

**先决条件**

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。

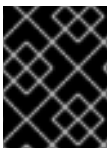
**流程**

- 使用 **ipa group-add-member** 命令向用户组添加成员。
  使用这些选项指定成员类型：
  - **--users** 添加 IdM 用户
  - **--external** 添加一个存在于 IdM 域外的用户，格式为 **DOMAIN\user_name** 或 **user_name@domain**
  - **--groups** 添加 IdM 用户组

  例如，将 group_b 添加为 group_a 的成员：

```
$ ipa group-add-member group_a --groups=group_b
Group name: group_a
GID: 1133400009
Member users: user_a
Member groups: group_b
Indirect Member users: user_b
-------------------------
Number of members added 1
-------------------------
```

  group_b 的成员现在是 group_a 的间接成员。



**重要**

将组添加为另一个组的成员时，请勿创建递归组。例如，如果组 A 是组 B 的成员，则不要将组 B 添加为组 A 的成员。递归组可能会导致无法预料的行为。

**注意**

将成员添加到用户组后，更新可能需要一些时间才能传播到身份管理环境中的所有客户端。这是因为，当任何给定主机解析用户、组和网络组时，**系统安全服务守护进程** (SSSD) 首先检查其缓存，并且仅对丢失或过期的记录执行服务器查找。

## 14.7. 添加没有用户私有组的用户

默认情况下，每当在 IdM 中创建新用户时，IdM 都会创建用户私有组(UPG)。UPG 是特定的组类型：

- UPG 与新创建的用户具有相同的名称。
- 用户是 UPG 的唯一成员。UPG 不能包含任何其他成员。
- 私有组的 GID 与用户的 UID 相匹配。

不过，可以添加用户而不创建 UPG 。

### 14.7.1. 没有用户私有组的用户

如果 NIS 组或其他系统组已使用将要分配给用户私有组的 GID，则有必要避免创建 UPG。

您可以通过两种方式执行此操作：

- 添加没有 UPG 的新用户，而不全局禁用私有组。请参阅 全局启用私有组时添加没有用户私有组的用户 。
- 对所有用户全局禁用 UPG，然后添加新用户。请参阅 对所有用户全局禁用用户私有组，和 在用户私有组全局禁用时添加用户。

在这两种情况下，在添加新用户时，IdM 都需要指定 GID，否则操作将失败。这是因为对于新用户，IdM 需要 GID，但默认用户组 **ipausers** 是一个非 POSIX 组，因此没有关联的 GID。您指定的 GID 不必对应于已经存在的组。

**注意**

指定 GID 不会创建新组。它仅为新用户设置 GID 属性，因为 IdM 需要属性。

### 14.7.2. 在全局启用私有组时添加没有用户私有组的用户

您可以添加用户而不创建用户私有组(UPG)，即使系统上启用了 UPG。这需要为新用户手动设置 GID。有关为何需要此功能的详情，请查看 没有用户私有组的用户。

**流程**

- 要防止 IdM 创建 UPG，请在 **ipa user-add** 命令中添加 **--noprivate** 选项。
  请注意，若要命令成功，您必须指定一个自定义的 GID。例如，使用 GID 10000 添加新用户：

```
$ ipa user-add jsmith --first=John --last=Smith --noprivate --gid 10000
```

### 14.7.3. 对所有用户全局禁用用户私有组

您可以在全局范围内禁用用户私有组(UPG)。这样可防止为所有新用户创建 UPG。现有用户不会受到这一更改的影响。

**流程**

1. 获取管理员权限：

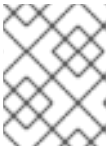
```
$ kinit admin
```

2. IdM 使用目录服务器管理的条目插件来管理 UPG。列出插件的实例：

```
$ ipa-managed-entries --list
```

3. 要确保 IdM 不创建 UPG，请禁用负责管理用户私有组的插件实例：

```
$ ipa-managed-entries -e "UPG Definition" disable
Disabling Plugin
```



**注意**

要在稍后重新启用 **UPG Definition** 实例，请使用 **ipa-managed-entries -e "UPG Definition" enable** 命令。

4. 重新启动目录服务器来加载新配置。

```
$ sudo systemctl restart dirsrv.target
```

要在禁用 UPG 后添加用户，您需要指定 GID。如需更多信息，请参阅在用户私有组群全局禁用时添加用户

**验证步骤**

- 要检查 UPG 是否全局禁用，请再次使用 disable 命令：

```
$ ipa-managed-entries -e "UPG Definition" disable
Plugin already disabled
```

### 14.7.4. 当全局禁用用户私有组时添加用户

当全局禁用用户私有组(UPG)时，IdM 不会自动为新用户分配 GID。要成功添加用户，您必须手动分配 GID，或使用自动成员规则来分配 GID。有关为何需要此功能的详情，请查看 没有用户私有组的用户。

**先决条件**

- 必须对所有用户全局禁用 UPG。如需更多信息，请参阅对所有用户全局禁用用户私有组

**流程**

- 要确保在禁用创建 UPG 时成功添加新用户，请选择以下之一：
  - 添加新用户时指定自定义的 GID。GID 不必对应于已经存在的用户组。
    例如，当从命令行添加用户时，请在 **ipa user-add** 命令中添加 **--gid** 选项。
  - 使用自动成员规则将用户添加到具有 GID 的现有组中。

## 14.8. 使用 IDM CLI 将用户或组作为成员管理者添加到 IDM 用户组中

按照以下流程，使用 IdM CLI 将用户或组作为成员管理者添加到 IdM 用户组。成员管理者可以将用户或组添加到 IdM 用户组中，但不能更改组的属性。

**先决条件**

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。
- 您必须具有要添加为成员管理器的用户名以及您要管理的组的名称。

**流程**

- 使用 **ipa group-add-member-manager** 命令，将用户作为成员管理者添加到 IdM 用户组。
  例如，要将用户 **test** 添加为 **group_a** 的成员管理者：

```
$ ipa group-add-member-manager group_a --users=test
Group name: group_a
GID: 1133400009
Membership managed by users: test
-------------------------
Number of members added 1
-------------------------
```

用户 **test** 现在可以管理 **group_a** 的成员。

- 使用 **ipa group-add-member-manager** 命令，将组作为成员管理者添加到 IdM 用户组。例如，要将 **group_admins** 添加为 **group_a** 的成员管理者：

```
$ ipa group-add-member-manager group_a --groups=group_admins
Group name: group_a
GID: 1133400009
Membership managed by groups: group_admins
Membership managed by users: test
-------------------------
Number of members added 1
-------------------------
```

组 **group_admins** 现在可以管理 **group_a** 的成员。

**注意**

将成员管理者添加到用户组后，可能需要过些时间，才能将更新传播到身份管理环境中的所有客户端。

**验证步骤**

- 使用 **ipa group-show** 命令来验证用户和组是否已被添加为成员管理者。

```
$ ipa group-show group_a
Group name: group_a
GID: 1133400009
Membership managed by groups: group_admins
Membership managed by users: test
```

**其它资源**

- 如需了解更多详细信息，请参阅 **ipa group-add-member-manager --help**。

## 14.9. 使用 IDM CLI 查看组成员

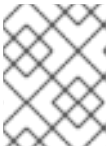
按照以下流程，使用 IdM CLI 查看组成员。您可以查看直接和间接组成员。如需更多信息，请参阅 直接和间接组成员。

**流程：**

- 要列出组成员，请使用 **ipa group-show *group_name*** 命令。例如：

```
$ ipa group-show group_a
```

```
...
Member users: user_a
Member groups: group_b
Indirect Member users: user_b
```



**注意**

间接成员列表不包括来自可信活动目录域的外部用户。活动目录信任用户对象在身份管理界面中不可见，因为它们在身份管理中不作为 LDAP 对象存在。

## 14.10. 使用 IDM CLI 从用户组中删除成员

按照以下流程，使用 IdM CLI 从用户组中删除成员。

**先决条件**

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. *可选*。使用 **ipa group-show** 命令确认组是否包含您要删除的成员。
2. 使用 **ipa group-remove-member** 命令从用户组中删除成员。
   使用这些选项来指定要删除的成员：
   - **--users** 删除 IdM 用户
   - **--external** 删除存在于 IdM 域外的用户，格式为 **DOMAIN\user_name** 或 **user_name@domain**
   - **--groups** 删除 IdM 用户组

   例如，要从名为 *group_name* 的组中删除 *user1*、*user2* 和 *group1*：

```
$ ipa group-remove-member group_name --users=user1 --users=user2 --groups=group1
```

## 14.11. 使用 IDM CLI 从 IDM 用户组中删除作为成员管理者的用户或组

按照以下流程，使用 IdM CLI，以成员管理者身份从 IdM 用户组中删除用户或组。成员管理者可以从 IdM 用户组中删除用户或组，但不能更改组的属性。

**先决条件**

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。
- 您必须具有要删除的现有成员管理者用户或组的名称，以及它们要管理的组的名称。

**流程**

- 使用 **ipa group-remove-member-manager** 命令，删除作为 IdM 用户组的成员管理者的用户。
  例如，要删除作为 **group_a** 的成员管理者的用户 **test** ：

```
$ ipa group-remove-member-manager group_a --users=test
```

```
Group name: group_a
GID: 1133400009
Membership managed by groups: group_admins
-------------------------
Number of members removed 1
-------------------------
```

用户 **test** 不再管理 **group_a** 的成员。

- 使用 **ipa group-remove-member-manager** 命令，删除作为 IdM 用户组的成员管理者的组。例如，要删除作为 **group_a** 的成员管理者的组 **group_admins** ：

```
$ ipa group-remove-member-manager group_a --groups=group_admins
Group name: group_a
GID: 1133400009
-------------------------
Number of members removed 1
-------------------------
```

组 **group_admins** 不再管理 **group_a** 的成员。

**注意**

从用户组中删除成员管理者后，可能需要稍等片刻才能将更新传播到身份管理环境中的所有客户端。

**验证步骤**

- 使用 **ipa group-show** 命令来验证用户和组是否已作为成员管理者被删除。

```
$ ipa group-show group_a
Group name: group_a
GID: 1133400009
```

**其它资源**

- 如需了解更多详细信息，请参阅 **ipa group-remove-member-manager --help**。

## 14.12. 为 IDM 中的本地和远程组启用组合并

组可以是集中管理的，由域，如身份管理(IdM)或活动(AD)提供，或者它们在本地系统上的 **etc/group** 文件中管理。在大多数情况下，用户依赖于集中管理的存储。然而，在某些情况下，软件仍依赖于已知组中的成员资格来管理访问控制。

如果要管理域控制器和本地 **etc/group** 中的文件组，您可以启用组合并。您可以配置 **nsswitch.conf** 文件，来检查本地文件和远程服务。如果组在这两个地方都出现，则将合并成员用户列表，并在单个响应中返回。

以下步骤描述了如何为用户 *idmuser* 启用组合并。

**流程**

1. 将 **[SUCCESS=merge]** 添加到 **/etc/nsswitch.conf** 文件中：

```
# Allow initgroups to default to the setting for group.
initgroups: sss [SUCCESS=merge] files
```

2. 将 *idmuser* 添加到 IdM 中：

```
# ipa user-add idmuser
First name: idm
Last name: user
---------------------
Added user "idmuser"
---------------------
User login: idmuser
First name: idm
Last name: user
Full name: idm user
Display name: idm user
Initials: tu
Home directory: /home/idmuser
GECOS: idm user
Login shell: /bin/sh
Principal name: idmuser@IPA.TEST
Principal alias: idmuser@IPA.TEST
Email address: idmuser@ipa.test
UID: 19000024
GID: 19000024
Password: False
Member of groups: ipausers
Kerberos keys available: False
```

3. 验证本地 **audio** 组的 GID。

```
$ getent group audio
---------------------
audio:x:63
```

4. 将组 **audio** 添加到 IdM 中：

```
$ ipa group-add audio --gid 63
-------------------
Added group "audio"
-------------------
Group name: audio
GID: 63
```

**注意**

您在将 **audio** 组添加到 IdM 时定义的 GID 必须与本地 **audio** 组的 GID 相同。

5. 将 *idmuser* 用户添加到 IdM **audio** 组中：

```
$ ipa group-add-member audio --users=idmuser
Group name: audio
GID: 63
```

```
Member users: idmuser
-------------------------
Number of members added 1
-------------------------
```

**验证**

1. 以 *idmuser* 身份登录。
2. 验证 *idmuser* 在其会话中是否有本地组：

```
$ id idmuser
uid=1867800003(idmuser) gid=1867800003(idmuser)
groups=1867800003(idmuser),63(audio),10(wheel)
```

## 14.13. 使用 ANSIBLE 为用户提供 ID 覆盖对 IDM 客户端上本地声音卡的访问权限

您可以使用 **ansible-freeipa 组和 idoverrideuser** 模块在 IdM 客户端上使身份管理(IdM)或 Active Directory (AD)用户成员。**这会授予 IdM 或 AD 用户对主机上声音卡的特权访问权限。该流程使用 Default Trust View ID 视图的示例，在第一个 playbook 任务中添加 aduser@addomain.com ID 覆盖。在下一个 playbook 任务中，在 IdM 中创建 音频 组，GID 为 63，它对应于 RHEL 主机上的本地 音频 组的 GID。同时，aduser@addomain.com ID 覆盖作为成员添加到 IdM 音频组中。**

**先决条件**

- **您有访问要在其上执行流程第一部分的 IdM 客户端的 root 访问权限。在示例中，这是 client.idm.example.com。**
- **您已配置了 Ansible 控制节点以满足以下要求：**
  - **您使用 Ansible 版本 2.14 或更高版本。**
  - **您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。**
  - **您使用 RHEL 8.10 或更高版本。**
  - **示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。**

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- AD 林与 IdM 信任。在示例中，AD 域的名称是 addomain.com，以及 AD 用户的完全限定域名(FQDN)，其存在于本地 音频 组中存在是 aduser@addomain.com。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 在 client.idm.example.com 上，将 [SUCCESS=merge] 添加到 /etc/nsswitch.conf 文件中：

```
[...]
# Allow initgroups to default to the setting for group.
initgroups: sss [SUCCESS=merge] files
```

2. 确定本地 音频 组的 GID：

```
$ getent group audio
--------------------
audio:x:63
```

3. 在 Ansible 控制节点上，创建一个带有任务的 add-aduser-to-audio-group.yml playbook，将 aduser@addomain.com 用户覆盖添加到 Default Trust View 中：

```
---
- name: Playbook to manage idoverrideuser
  hosts: ipaserver
  become: false

  tasks:
  - name: Add aduser@addomain.com user to the Default Trust View
    ipaidoverrideuser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      idview: "Default Trust View"
      anchor: aduser@addomain.com
```

4. 在同一 playbook 中使用另一个 playbook 任务，将组 音频 添加到 IdM 中，GID 为 63。将 aduser idoverrideuser 添加到组中：

```
- name: Add the audio group with the aduser member and GID of 63
  ipagroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: audio
    idoverrideuser:
    - aduser@addomain.com
    gidnumber: 63
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-aduser-to-audio-group.yml
```

验证

1. 以 AD 用户身份登录 IdM 客户端：

```
$ ssh aduser@addomain.com@client.idm.example.com
```

2. 验证 AD 用户的组成员资格：

```
$ id aduser@addomain.com
uid=702801456(aduser@addomain.com) gid=63(audio) groups=63(audio)
```

**其它资源**

- idoverrideuser 和 ipagroup ansible-freeipa 上游文档
- 为 IdM 中的本地和远程组启用组合并

# 第 15 章 在 IDM WEB UI 中管理用户组

本章介绍了使用 IdM Web UI 的用户组管理。

用户组是一组具有常见特权、密码策略和其他特征的用户。

Identity Management (IdM) 中的用户组可以包括：

- IdM 用户
- 其他 IdM 用户组
- 外部用户，即 IdM 之外的用户

## 15.1. IDM 中的不同组类型

IdM 支持以下类型的组：

**POSIX 组（默认）**

POSIX 组支持其成员的 Linux POSIX 属性。请注意，与 Active Directory 交互的组无法使用 POSIX 属性。

POSIX 属性将用户识别为单独的实体。与用户相关的 POSIX 属性示例包括 uidNumber（一个用户号 (UID)）和 gidNumber（一个组号 (GID)）。

**非 POSIX 组**

非 POSIX 组不支持 POSIX 属性。例如，这些组没有定义 GID。

这种组的所有成员必须属于 IdM 域。

**外部组**

使用外部组添加存在于 IdM 域外部的身份存储中的组成员，例如：

- 本地系统
- Active Directory 域
- 目录服务

外部组不支持 POSIX 属性。例如，这些组没有定义 GID。

表 15.1. 默认创建的用户组

| 组名称 | 默认组成员 |
| --- | --- |
| **ipausers** | 所有 IdM 用户 |
| **admins** | 具有管理特权的用户，包括默认的 **admin** 用户 |
| **editors** | 这是一个旧的组，不再具有任何特殊权限 |
| **trust admins** | 具有管理 Active Directory 信任权限的用户 |

将用户添加到用户组时，该用户将获得与组关联的特权和策略。例如，若要向用户授予管理特权，可将该用户添加到 **admins** 组。

**警告**

不要删除 **admins** 组。由于 **admins** 是 IdM 要求的预定义组，因此此操作会导致某些命令出现问题。

另外，当在 IdM 中创建新用户时，IdM 默认会创建*用户私有组*。有关私有组的更多信息，请参阅在没有私有组的情况下添加用户。

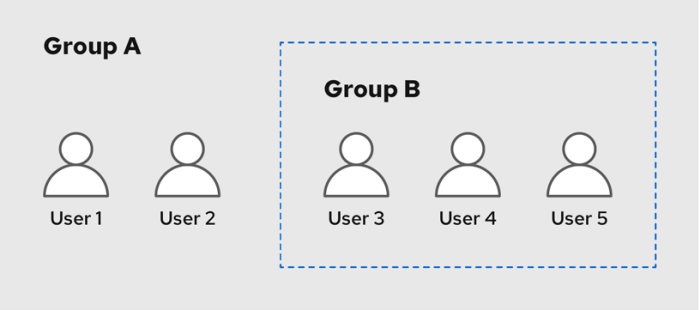
## 15.2. 直接和间接组成员

IdM 中的用户组属性适用于直接和间接成员：当组 B 是组 A 的成员时，组 B 中的所有用户都被视为组 A 的间接成员。

例如，在下图中：

- 用户 1 和用户 2 是组 A 的*直接成员*。
- 用户 3、用户 4 和用户 5 是组 A 的*间接成员*。

图 15.1. 直接和间接组成员身份




如果您为用户组 A 设置密码策略，该策略也适用于用户组 B 中的所有用户。

## 15.3. 使用 IDM WEB UI 添加用户组

按照以下流程，使用 IdM Web UI 添加用户组。

**先决条件**

- 已登陆到 IdM Web UI。

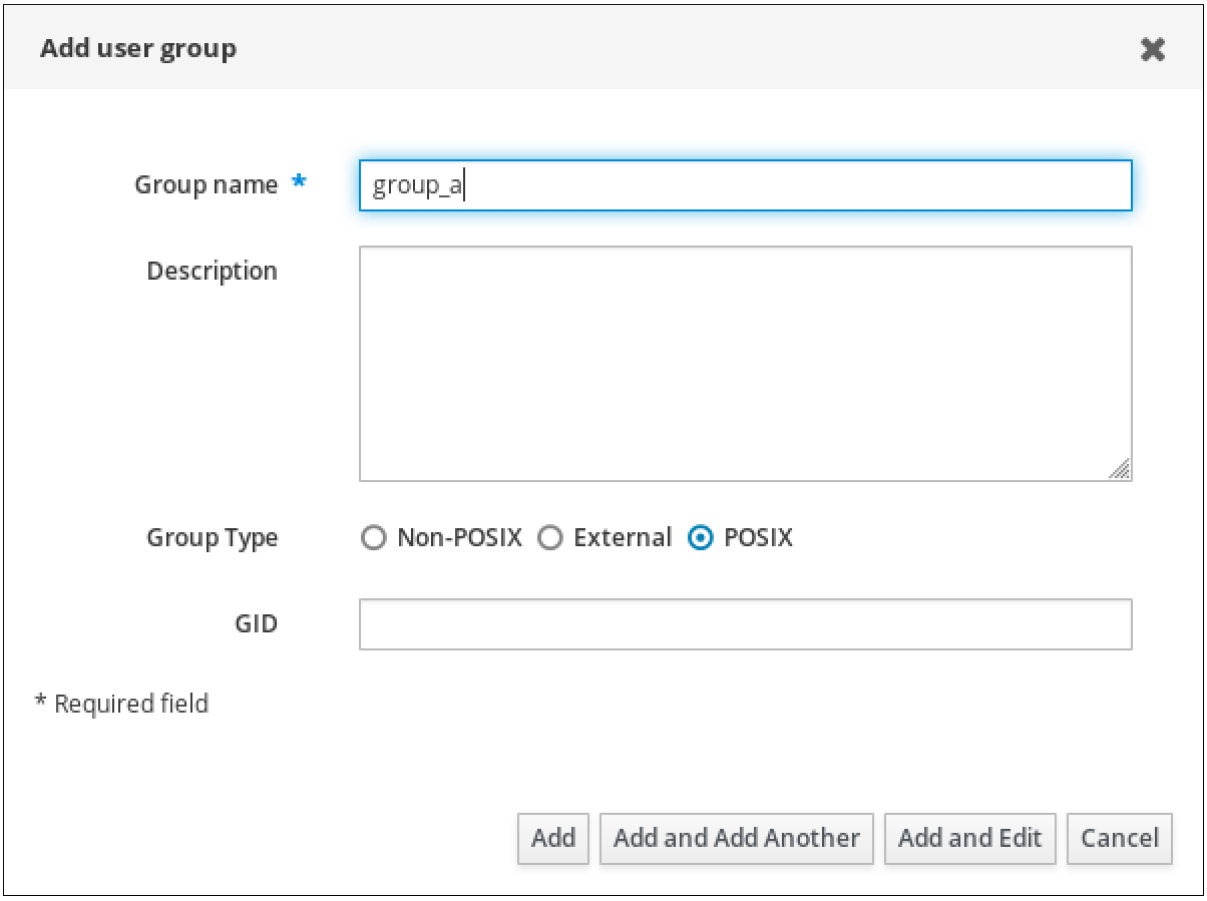
**流程**

1. 点击 Identity → Groups，然后选择左侧栏中的 User Groups。

2. 单击 **Add** 开始添加组。

3. 填写有关组的信息。有关用户组类型的更多信息，请参阅 **IdM 中不同的组类型**。

您可以为组指定自定义的 GID。如果您这样做，请小心以避免 ID 冲突。如果没有指定自定义的 GID，IdM 会自动从可用的 ID 范围内分配一个 GID。




4. 单击 **Add** 确认。

## 15.4. 使用 IDM WEB UI 删除用户组

按照以下流程，使用 IdM Web UI 删除用户组。请注意，删除组不会从 IdM 中删除组成员。

**先决条件**

- 已登陆到 IdM Web UI。

**流程**

1. 点击 **Identity → Groups**，并选择 **User Groups**。
2. 选择要删除的组。
3. 单击 **Delete**。
4. 单击 **Delete** 确认。

### 15.5. 使用 IDM WEB UI 将成员添加到用户组中

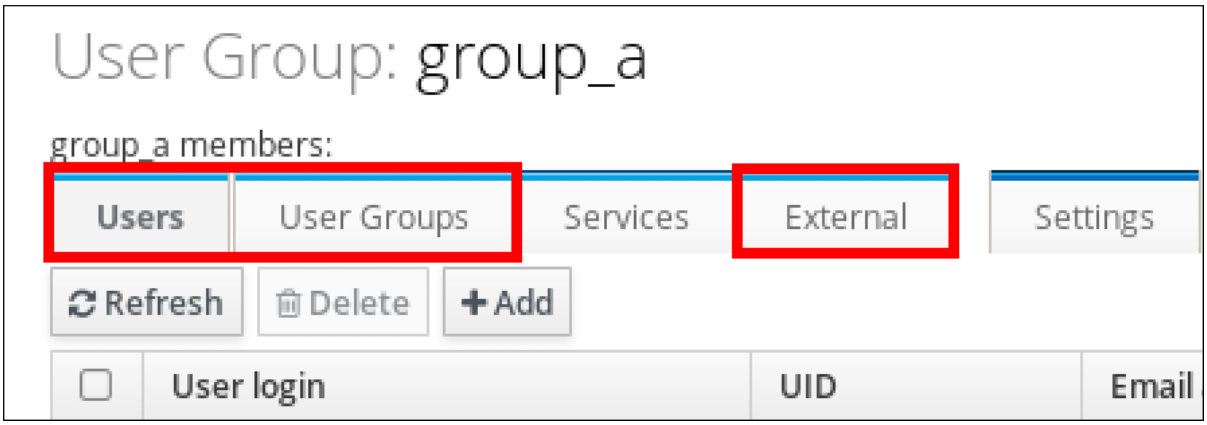
您可以将用户和用户组添加为用户组的成员。如需更多信息，请参阅 IdM 中不同的组类型 和 直接和间接组成员 。

**先决条件**

- 已登陆到 IdM Web UI。

**流程**

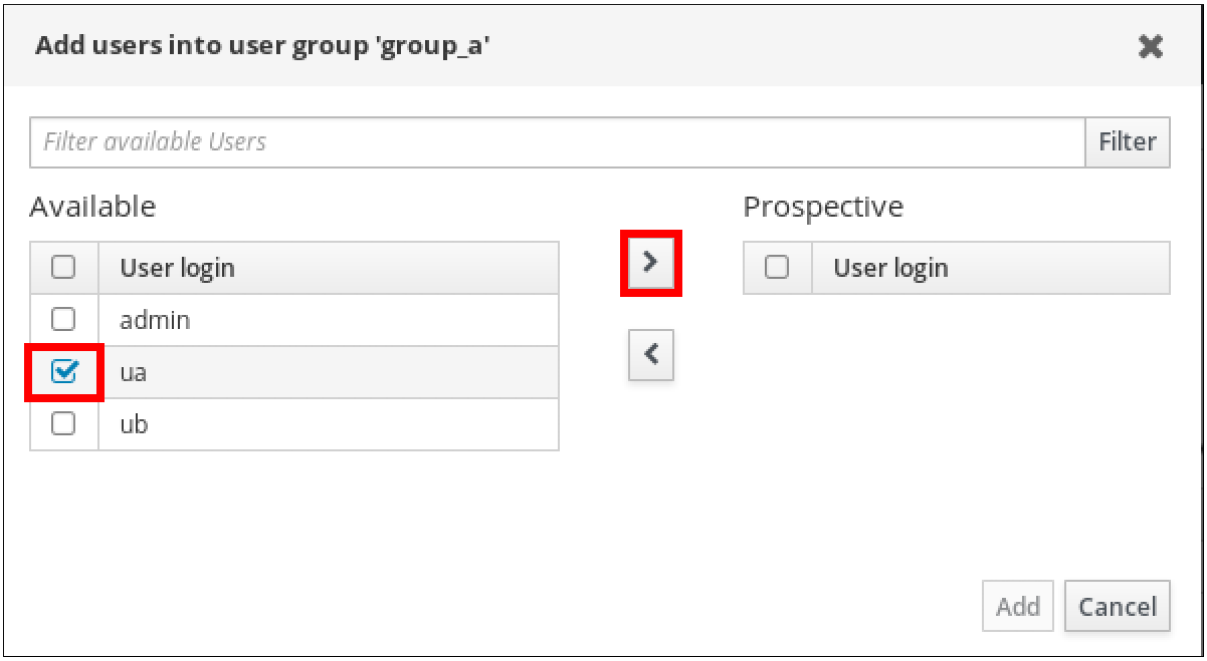
1. 单击 **Identity → Groups**，并选择左侧栏中的 **User Groups**。
2. 单击组的名称。
3. 选择您要添加的组成员的类型： **User**、**User Groups** 或 **External**。



4. 点击 **Add**。

5. 选中您要添加的一个或多个成员旁边的复选框。

6. 单击向右箭头，将选定的成员移到组中。




7. 单击 **Add** 确认。

## 15.6. 使用 WEB UI 将用户或组作为成员管理者添加到 IDM 用户组中

按照以下流程，使用 Web UI，以成员管理者身份将用户或组添加到 IdM 用户组。成员管理者可以将用户或组添加到 IdM 用户组中，但不能更改组的属性。

**先决条件**

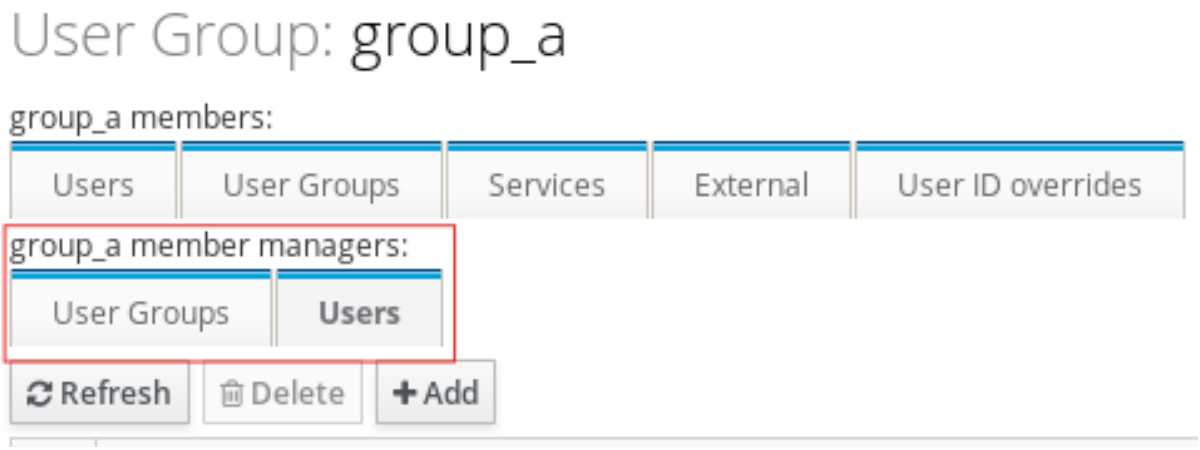
- 已登陆到 IdM Web UI。
- 您必须具有要添加为成员管理器的用户名以及您要管理的组的名称。

**流程**

1. 单击 **Identity → Groups**，并选择左侧栏中的 **User Groups**。

2. 单击组的名称。

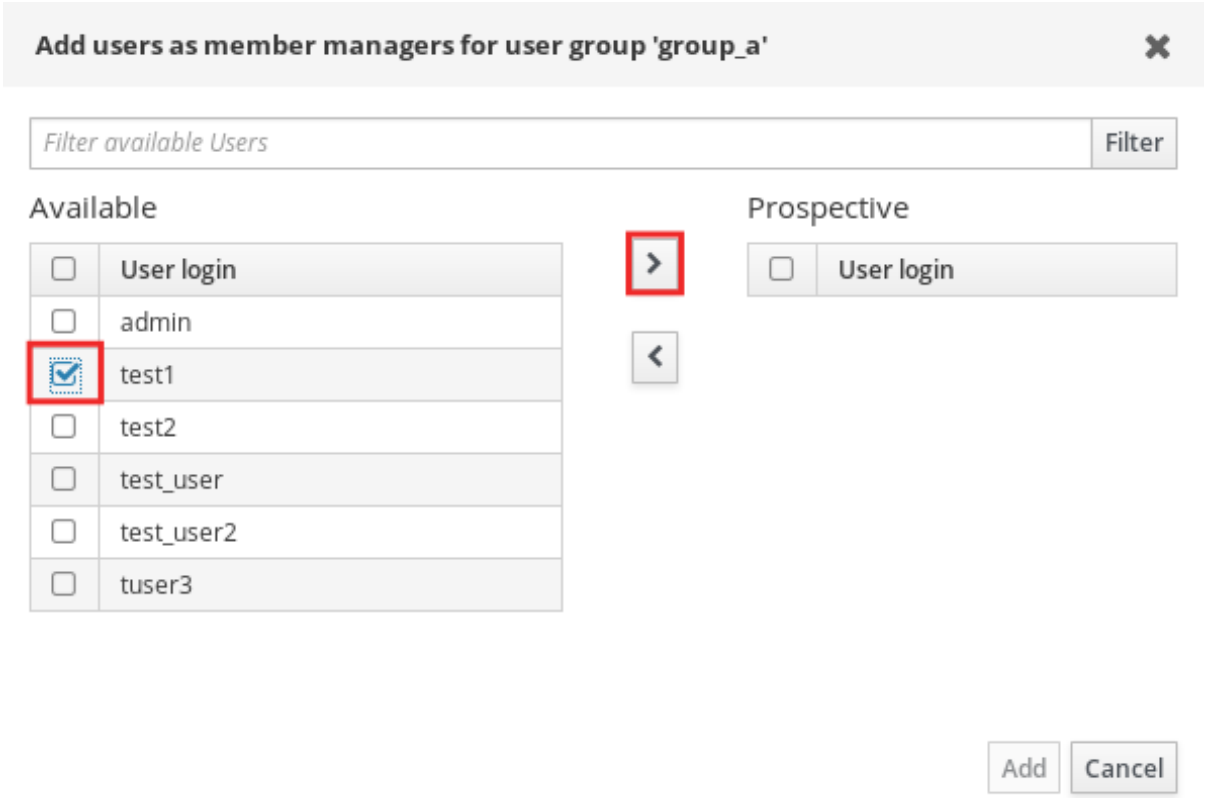
3. 选择您要添加的组成员管理者的类型：**Users** 或 **User Groups**。



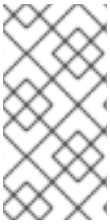

4. 点击 **Add**。

5. 选中您要添加的一个或多个成员旁边的复选框。

6. 单击向右箭头，将选定的成员移到组中。




7. 单击 **Add** 确认。

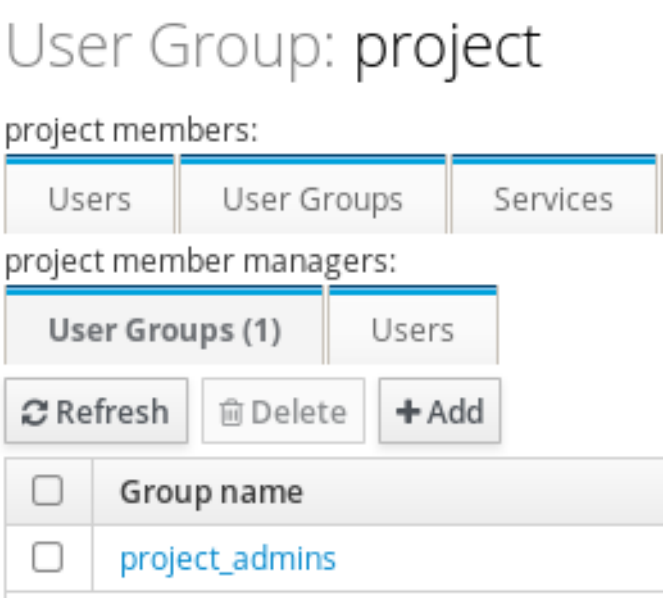


**注意**

**将成员管理者添加到用户组后，可能需要过些时间，才能将更新传播到身份管理环境中的所有客户端。**

**验证步骤**

- **验证新添加的用户或用户组是否已添加到用户或用户组的成员管理者列表中：**




**其它资源**

- 如需更多信息，请参阅 **ipa group-add-member-manager --help**。

### 15.7. 使用 IDM WEB UI 查看组成员

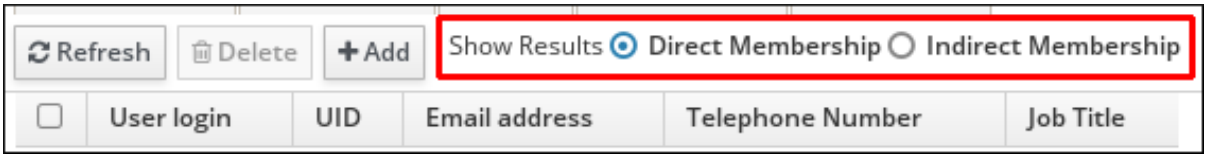
按照以下流程，使用 IdM Web UI 查看组成员。您可以查看直接和间接组成员。如需更多信息，请参阅直接和间接组成员。

**先决条件**

- 已登陆到 IdM Web UI。

**流程**

1. 选择 **Identity** → **Groups**。
2. 在左侧栏中选择 **User Groups** 。
3. 单击您要查看的组的名称。
4. 在 **Direct Membership** 和 **Indirect Membership** 之间切换。




### 15.8. 使用 IDM WEB UI 从用户组中删除成员

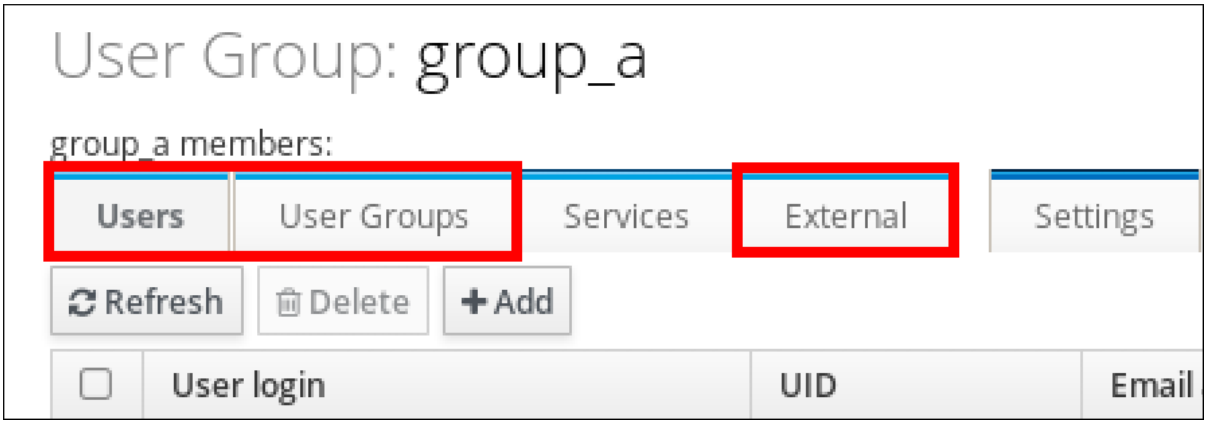
**按照以下流程，使用 IdM Web UI 从用户组中删除成员。**

**先决条件**

- **已登陆到 IdM Web UI。**

**流程**

1. **单击 Identity → Groups，并选择左侧栏中的 User Groups。**
2. **单击组的名称。**
3. **选择要删除的组成员的类型： User、User Groups 或 External。**




4. **选中您要删除的成员旁边的复选框。**
5. **单击 Delete。**
6. **单击 Delete 确认。**

## 15.9. 使用 WEB UI 从 IDM 用户组中删除作为成员管理者的用户或组

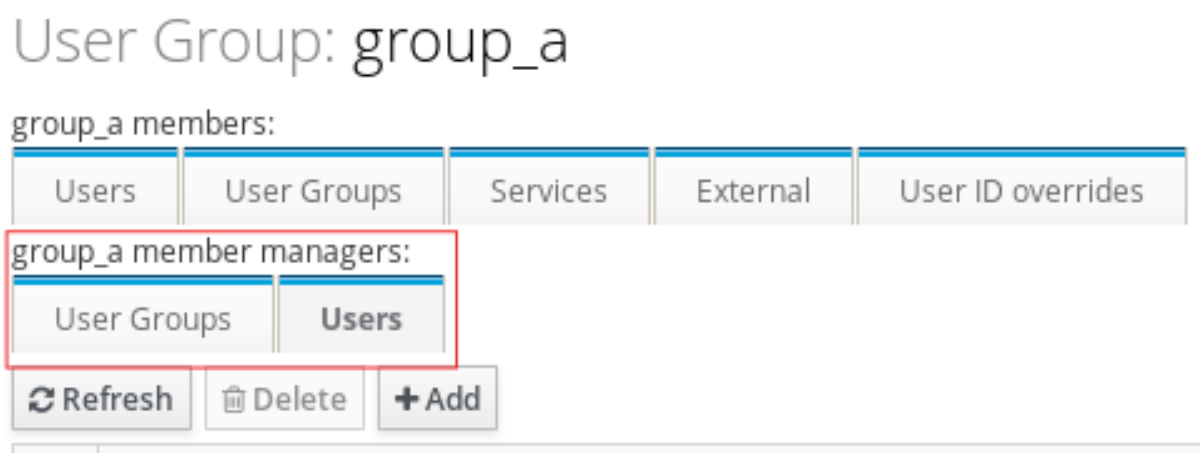
按照以下流程，使用 Web UI，以成员管理者的身份从 IdM 用户组中删除用户或组。成员管理者可以从 IdM 用户组中删除用户或组，但不能更改组的属性。

### 先决条件

- 已登陆到 IdM Web UI。
- 您必须具有要删除的现有成员管理者用户或组的名称，以及它们要管理的组的名称。

### 流程

1. 单击 **Identity → Groups**，并选择左侧栏中的 **User Groups**。
2. 单击组的名称。
3. 选择您要删除的成员管理者的类型： **Users** 或 **User Groups**。



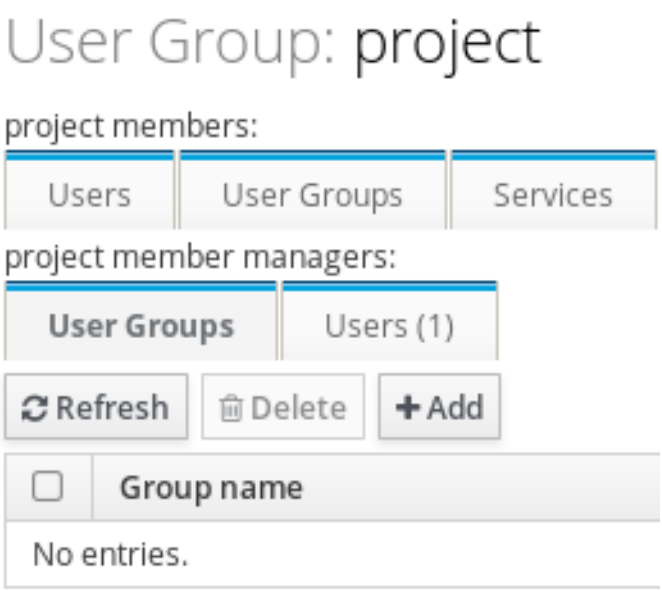

4. 选中您要删除的成员管理者旁边的复选框。
5. 单击 **Delete**。
6. 单击 **Delete** 确认。

**注意**

从用户组中删除成员管理者后，可能需要稍等片刻才能将更新传播到身份管理环境中的所有客户端。

**验证步骤**

- 验证用户或用户组是否已从用户或用户组的成员管理者列表中删除：




**其它资源**

- 如需了解更多详细信息，请参阅 ipa group-add-member-manager --help。

# 第 16 章 使用 ANSIBLE PLAYBOOK 管理用户组

本节介绍使用 Ansible playbook 进行用户组管理。

用户组是一组具有常见特权、密码策略和其他特征的用户。

Identity Management (IdM) 中的用户组可以包括：

- IdM 用户
- 其他 IdM 用户组
- 外部用户，即 IdM 之外的用户

本节包括以下主题：

- IdM 中的不同组类型
- 直接和间接组成员
- 使用 Ansible playbook 确保存在 IdM 组和组成员
- 使用 Ansible 启用 AD 用户管理 IdM
- 使用 Ansible playbook 在 IDM 用户组中存在成员管理器
- 使用 Ansible playbook，确保 IDM 用户组中没有成员管理器

## 16.1. IDM 中的不同组类型

IdM 支持以下类型的组：

POSIX 组（默认）

POSIX 组支持其成员的 Linux POSIX 属性。请注意，与 Active Directory 交互的组无法使用 POSIX 属性。

POSIX 属性将用户识别为单独的实体。与用户相关的 POSIX 属性示例包括 uidNumber（一个用户号 (UID)）和 gidNumber（一个组号 (GID)）。

非 POSIX 组

非 POSIX 组不支持 POSIX 属性。例如，这些组没有定义 GID。

这种组的所有成员必须属于 IdM 域。

外部组

使用外部组添加存在于 IdM 域外部的身份存储中的组成员，例如：

- 本地系统
- Active Directory 域
- 目录服务

外部组不支持 POSIX 属性。例如，这些组没有定义 GID。

表 16.1. 默认创建的用户组

| 组名称 | 默认组成员 |
| --- | --- |
| **ipausers** | 所有 IdM 用户 |
| **admins** | 具有管理特权的用户，包括默认的 **admin** 用户 |
| **editors** | 这是一个旧的组，不再具有任何特殊权限 |

| 组名称 | 默认组成员 |
| --- | --- |
| **trust admins** | 具有管理 Active Directory 信任权限的用户 |

将用户添加到用户组时，该用户将获得与组关联的特权和策略。例如，若要向用户授予管理特权，可将该用户添加到 admins 组。

**警告**

不要删除 admins 组。由于 admins 是 IdM 要求的预定义组，因此此操作会导致某些命令出现问题。

另外，当在 IdM 中创建新用户时，IdM 默认会创建*用户私有组*。有关私有组的更多信息，请参阅在没有私有组的情况下添加用户。

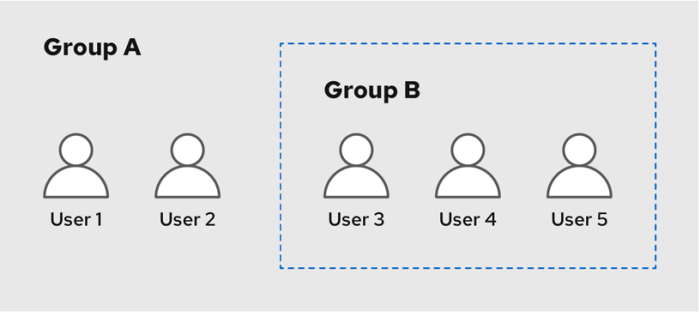
## 16.2. 直接和间接组成员

IdM 中的用户组属性适用于直接和间接成员：当组 B 是组 A 的成员时，组 B 中的所有用户都被视为组 A 的间接成员。

例如，在下图中：

- 用户 1 和用户 2 是组 A 的*直接成员*。
- 用户 3、用户 4 和用户 5 是组 A 的*间接成员*。

图 16.1. 直接和间接组成员身份




如果您为用户组 A 设置密码策略，该策略也适用于用户组 B 中的所有用户。

## 16.3. 使用 ANSIBLE PLAYBOOK 确保存在 IDM 组和组成员

以下流程描述了使用 Ansible playbook 确保存在 IdM 组和组成员（用户和用户组）。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
-

IdM 中已存在您想要引用的用户。有关确保存在使用 Ansible 的用户的详细信息，请参阅使用 Ansible playbook 管理用户帐户。

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的用户和组信息创建一个 Ansible playbook 文件：

```
---
- name: Playbook to handle groups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Create group ops with gid 1234
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: ops
      gidnumber: 1234

  - name: Create group sysops
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: sysops
      user:
      - idm_user

  - name: Create group appops
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: appops

  - name: Add group members sysops and appops to group ops
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: ops
      group:
      - sysops
      - appops
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-group-
members.yml
```

验证步骤

您可以使用 ipa group-show 命令验证 ops 组是否包含 sysops 和 appops 作为直接成员，idm_user 作为间接成员：

1. 以管理员身份登录到 ipaserver：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 显示关于 *ops* 的信息：

```
ipaserver]$ ipa group-show ops
  Group name: ops
  GID: 1234
  Member groups: sysops, appops
  Indirect Member users: idm_user
```

IdM 中已存在 appops 和 sysops 组，后者包括 idm_user 用户。

其它资源

- 请参阅 /usr/share/doc/ansible-freeipa/README-group.md Markdown 文件。

## 16.4. 使用 ANSIBLE 在单个任务中添加多个 IDM 组

您可以使用 ansible-freeipa ipagroup 模块，在单个 Ansible 任务中添加、修改和删除多个身份管理(IdM)用户组。为此，请使用 ipagroup 模块的 groups 选项。

使用 groups 选项，您还可以指定仅应用到特定组的多个组变量。根据 name 变量定义此组，这是 groups 选项的唯一强制变量。

完成此流程，以确保在单个任务中，在 IdM 中存在 sysops 和 appops 组。将 sysops 组定义为非 posix 组，将 appops 组定义为外部组。

**先决条件**

- 在控制节点上：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已安装 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您正在使用 RHEL 8.9 及更新版本。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。

**流程**

1. 使用以下内容创建 Ansible playbook 文件 add-nonposix-and-external-groups.yml ：

```
---
- name: Playbook to add nonposix and external groups
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Add nonposix group sysops and external group appops
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      groups:
      - name: sysops
        nonposix: true
      - name: appops
        external: true
```

2. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/add-nonposix-
and-external-groups.yml
```

其它资源

- ansible-freeipa 上游文档中的 group 模块

## 16.5. 使用 ANSIBLE 启用 AD 用户管理 IDM

按照以下流程，使用 Ansible playbook 确保户 ID 覆盖在身份管理(IdM)组中存在。用户 ID 覆盖是您在使用 AD 建立信任视图中创建的 Active Directory (AD)用户覆盖。因此，运行 playbook （如 AD 用户）能够完全管理 IdM，而无需两个不同的帐户和密码。

先决条件

- 您需要知道 IdM admin 密码。
- 已使用 AD 安装信任。
- IdM 中已存在 AD 用户的用户 ID 覆盖。如果没有，使用 ipa idoverrideuser-add 'default trust view' *ad_user@ad.example.com* 命令创建它。
- 您要将用户 ID 覆盖添加至其中的组在 IdM 中已存在。
- 您可以使用 IdM 的 4.8.7 版本或更高版本。要查看您在服务器上安装的 IdM 版本，请输入 ipa --version。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  -

示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 使用以下内容创建 add-useridoverride-to-group.yml playbook：

```
---
- name: Playbook to ensure presence of users in a group
  hosts: ipaserver


  - name: Ensure the ad_user@ad.example.com user ID override is a member of the
admins group:
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: admins
      idoverrideuser:
      - ad_user@ad.example.com
```

在示例中：

- Secret123 是 IdM 管理员密码。

- 管理员 是您要添加 ad_user@ad.example.com ID 覆盖的 IdM POSIX 组的名称。此组成员具有全部的管理员特权。

- ad_user@ad.example.com 是 AD 管理员的用户 ID 覆盖。用户存储在已建立信任的 AD 域中。

3. 保存该文件。

4. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-useridoverride-to-group.yml
```

其它资源

- AD 用户的 ID 覆盖
- /usr/share/doc/ansible-freeipa/README-group.md
- /usr/share/doc/ansible-freeipa/playbooks/user
- 在 Active Directory 环境中使用 ID 视图
- 启用 AD 用户管理 IdM

## 16.6. 使用 ANSIBLE PLAYBOOK 确保 IDM 用户组中存在成员管理器

以下流程描述了使用 Ansible playbook 确保存在 IdM 成员管理器（用户和用户组）。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。

- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 您必须具有要添加为成员管理器的用户名以及您要管理的组的名称。

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的用户和组成员管理信息创建一个 Ansible playbook 文件：

```
---
- name: Playbook to handle membership management
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure user test is present for group_a
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: group_a
      membermanager_user: test

  - name: Ensure group_admins is present for group_a
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: group_a
      membermanager_group: group_admins
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-member-
managers-user-groups.yml
```

**验证步骤**

您可以使用 ipa group-show 命令验证 group_a 组是否包含 test 作为成员管理者，以及 group_admins 为 group_a 的成员管理者：

1. 以管理员身份登录到 ipaserver：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 显示 *managergroup1* 的信息：

```
ipaserver]$ ipa group-show group_a
  Group name: group_a
  GID: 1133400009
  Membership managed by groups: group_admins
  Membership managed by users: test
```

**其它资源**

- 请参阅 ipa host-add-member-manager --help。
- 请参阅 ipa man page。

## 16.7. 使用 ANSIBLE PLAYBOOK 确保 IDM 用户组中没有成员管理者

以下流程描述了在使用 Ansible playbook 时确保 IdM 成员管理者（用户和用户组）不存在。

**先决条件**

- 您知道 IdM 管理员密码。

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您必须具有要删除的现有成员管理者用户或组的名称，以及它们要管理的组的名称。

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的用户和组成员管理信息创建一个 Ansible playbook 文件：

```
---
- name: Playbook to handle membership management
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure member manager user and group members are absent for group_a
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: group_a
      membermanager_user: test
```

```
    membermanager_group: group_admins
    action: member
    state: absent
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
member-managers-are-absent.yml
```

验证步骤

您可以使用 ipa group-show 命令验证 group_a 组不包含 test 作为成员管理者，以及 group_admins 为 group_a 的成员管理者：

1. 以管理员身份登录到 ipaserver：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 显示 group_a 的信息：

```
ipaserver]$ ipa group-show group_a
  Group name: group_a
  GID: 1133400009
```

其它资源

- 请参阅 ipa host-remove-member-manager --help。
- 请参阅 ipa man page。

# 第 17 章 使用 IDM CLI 自动化组成员资格

通过自动化组成员资格，您可以根据其属性自动将用户和主机分配到组。例如，您可以：

- 根据员工的经理、位置或任何其他属性，将员工的用户条目划分为组。
- 根据主机的类、位置或任何其他属性来划分主机。
- 将所有用户或全部主机添加到单个全局组。

本章涵盖了以下主题：

- 使用 IdM CLI 配置默认的自动成员组

## 17.1. 自动化组成员资格的好处

对用户使用自动成员资格，允许您：

- 减少手动管理组成员资格的开销

  您不再需要手动将每个用户和主机分配到组中。

- 提高用户和主机管理的一致性

  用户和主机根据严格定义的和自动评估的标准被分配到组。

- 简化基于组的设置的管理

  为组定义各种设置，然后应用到各个组成员，如 sudo 规则、自动挂载或访问控制。将用户和主机添加到组中会自动使管理这些设置变得更加简单。

## 17.2. 自动成员规则

在配置自动化组成员资格时，管理员定义自动成员规则。自动成员规则应用到特定的用户或主机目标组。它不能一次应用到多个组。

创建规则后，管理员会为其添加条件。它们指定将哪些用户或主机包含在目标组中，或从目标组中排除：

- 包含的条件

  当用户或主机条目满足包含的条件时，它将包含在目标组中。

- 排他条件

当用户或主机条目满足排他条件时，它不会包含在目标组中。

条件被指定为 Perl 兼容的正则表达式(PCRE)格式的正则表达式。有关 PCRE 的更多信息，请参阅 pcresyntax (3) 手册页。

注意

IdM 在包含条件之前评估排他条件。在发生冲突时，排他条件优先于包含条件。

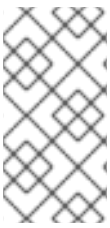
自动成员规则适用于将来创建的每个条目。这些条目将自动添加到指定的目标组中。如果一个条目满足多个自动成员规则中指定的条件，它将被添加到所有对应的组中。

现有条目 不会受到 新规则的影响。如果要更改现有条目，请参阅使用 IdM CLI 将自动成员规则应用到现有条目。

## 17.3. 使用 IDM CLI 添加自动成员规则

按照以下流程，使用 IdM CLI 添加自动成员规则。有关自动成员规则的详情，请参考 自动成员规则。

添加自动成员规则后，您可以在 向自动成员规则中添加条件 中所述的流程为其添加条件 。



注意

现有条目 不会受到 新规则的影响。如果要更改现有条目，请参阅使用 IdM CLI 将自动成员规则应用到现有条目。

**先决条件**

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。
- 新规则的目标组必须在 IdM 中存在。

**流程**

1. 输入 ipa automember-add 命令，来添加自动成员规则。

2. 在提示时，指定：

- 自动成员规则。这是目标组名称。
- 分组类型。这将指定规则以用户组还是主机组为目标。要以用户组为目标，请输入 group。要以主机组为目标，请输入 hostgroup。

例如，要为名为 user_group 的用户组添加自动成员规则：

```
$ ipa automember-add
Automember Rule: user_group
Grouping Type: group
-------------------------------
Added automember rule "user_group"
-------------------------------
  Automember Rule: user_group
```

验证步骤

- 您可以使用 使用 IdM CLI 查看现有的自动成员规则，来显示 IdM 中现有的自动成员资格规则和条件。

### 17.4. 使用 IDM CLI 将条件添加到自动成员规则中

配置自动成员规则后，您可以使用 IdM CLI 向该自动成员规则添加条件。有关自动成员规则的详情，请参考 自动成员规则。

先决条件

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。
- 目标规则必须在 IdM 中存在。详情请参阅 使用 IdM CLI 添加自动成员规则。

流程

1. 使用 ipa automember-add-condition 命令定义一个或多个包含或排他条件。

2. 在提示时，指定：

- 自动成员规则。这是目标规则名称。详情请查看 自动成员规则。
- 属性键。这将指定过滤器将应用到的条目属性。例如，用户的 uid ：
- 分组类型。这将指定规则以用户组还是主机组为目标。要以用户组为目标，请输入 group。要以主机组为目标，请输入 hostgroup。
- 包含正则表达式 和 排他正则表达式。它们将一个或多个条件指定为正则表达式。如果您只想指定一个条件，请在提示输入其它条件时按 Enter 键。

例如，以下条件针对用户登录属性(uid)中带有任何值(.*)的所有用户。

```
$ ipa automember-add-condition
Automember Rule: user_group
Attribute Key: uid
Grouping Type: group
[Inclusive Regex]: .*
[Exclusive Regex]:
--------------------------------
Added condition(s) to "user_group"
--------------------------------
  Automember Rule: user_group
  Inclusive Regex: uid=.*
----------------------------
Number of conditions added 1
----------------------------
```

再举一个例子，您可以使用自动成员资格规则以从活动目录(AD)中同步的所有 Windows 用户为目标。要达到此目的，请创建一个条件，该条件以其objectClass 属性中带有 ntUser 的用户为目标，该属性由所有 AD 用户共享：

```
$ ipa automember-add-condition
Automember Rule: ad_users
Attribute Key: objectclass
Grouping Type: group
[Inclusive Regex]: ntUser
```

```
[Exclusive Regex]:
-------------------------------------
Added condition(s) to "ad_users"
-------------------------------------
  Automember Rule: ad_users
  Inclusive Regex: objectclass=ntUser
----------------------------
Number of conditions added 1
----------------------------
```

验证步骤

- 您可以使用 使用 IdM CLI 查看现有的自动成员规则，来显示 IdM 中现有的自动成员资格规则和条件。

## 17.5. 使用 IDM CLI 查看现有的自动成员规则

按照以下流程，使用 IdM CLI 查看现有的自动成员规则。

**先决条件**

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. 输入 ipa automember-find 命令。
2. 在提示时，指定 Grouping type ：
   - 要以用户组为目标，请输入 group。
   - 要以主机组为目标，请输入 hostgroup。

   例如：

```
$ ipa automember-find
Grouping Type: group
---------------
1 rules matched
```

```
---------------
 Automember Rule: user_group
 Inclusive Regex: uid=.*
----------------------------
Number of entries returned 1
----------------------------
```

## 17.6. 使用 IDM CLI 删除自动成员规则

按照以下流程，使用 IdM CLI 删除自动成员规则。

删除自动成员规则也会删除与规则相关的所有条件。要只从规则中删除特定条件，请参阅 使用 IdM CLI 从自动成员规则中删除条件。

### 先决条件

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。

### 流程

1. 输入 ipa automember-del 命令。
2. 在提示时，指定：
   - 自动成员规则。这是您要删除的规则。
   - 分组规则。这将指定您要删除的规则是针对用户组的还是主机组的。输入 group 或 hostgroup 。

## 17.7. 使用 IDM CLI 从自动成员规则中删除条件

按照以下步流程，从自动成员规则中删除特定条件。

### 先决条件

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。

流程

1. 输入 ipa automember-remove-condition 命令。

2. 在提示时，指定：

- 自动成员规则。这是您要从中删除条件的规则的名称。
- 属性键。这是目标条目属性。例如，用户的 uid ：
- 分组类型。这将指定您要删除的条件是针对用户组的还是主机组的。输入 group 或 hostgroup 。
- 包含正则表达式 和 排他正则表达式。它们指定您要删除的条件。如果您只想指定一个条件，请在提示输入其它条件时按 Enter 键。

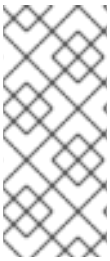
例如：

```
$ ipa automember-remove-condition
Automember Rule: user_group
Attribute Key: uid
Grouping Type: group
[Inclusive Regex]: .*
[Exclusive Regex]:
-------------------------------
Removed condition(s) from "user_group"
-------------------------------
  Automember Rule: user_group
-----------------------------
Number of conditions removed 1
-----------------------------
```

## 17.8. 使用 IDM CLI 将自动成员规则应用到现有条目

自动成员规则在规则添加后，自动应用到所创建的用户和主机条目。它们不会追溯到在规则添加之前存在的条目。

要将自动成员规则应用到之前添加的条目，您必须手动重建自动成员资格。重建自动成员资格会重新评估所有现有的自动成员规则，并将其应用到所有用户或主机条目或特定的条目。



**注意**

重建自动成员资格 不会 从组中删除用户或主机条目，即使条目不再与组的包含条件匹配。要手动删除它们，请参阅 使用 IdM CLI 从用户组中删除成员 或 使用 CLI 删除 IdM 主机组成员。

**先决条件**

- 您必须以管理员身份登录。详情请查看 link: 使用 kinit 手动登录到 IdM。

**流程**

- 要重建自动成员资格，请输入 ipa automember-rebuild 命令。使用以下选项指定要定为目标的条目：
  - 要为所有用户重建自动成员资格，请使用 --type=group 选项：

```
$ ipa automember-rebuild --type=group
---------------------------------------------------
Automember rebuild task finished. Processed (9) entries.
---------------------------------------------------
```

  - 要为所有主机重建自动成员资格，请使用 --type=hostgroup 选项。
  - 要为指定的一个用户或多个用户重建自动成员资格，请使用 --users=*target_user* 选项：

```
$ ipa automember-rebuild --users=target_user1 --users=target_user2
---------------------------------------------------
Automember rebuild task finished. Processed (2) entries.
---------------------------------------------------
```

  - 要为指定一个主机或多个主机重建自动成员资格，请使用 --hosts=*client.idm.example.com* 选项。

## 17.9. 使用 IDM CLI 配置默认的自动成员组

当您配置默认的自动成员组时，与任何自动成员规则不匹配的新用户或主机条目将自动添加到此默认组中。

**先决条件**

- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。
- 您要设置为默认的目标组在 IdM 中已存在。

**流程**

1. 输入 ipa automember-default-group-set 命令，来配置默认的自动成员组。
2. 在提示时，指定：
   - Default (fallback) Group，指定目标组名称。
   - Grouping Type，指定目标是用户组还是主机组。要以用户组为目标，请输入 group。要以主机组为目标，请输入 hostgroup。

     例如：

```
$ ipa automember-default-group-set
Default (fallback) Group: default_user_group
Grouping Type: group
-------------------------------------------------
Set default (fallback) group for automember "default_user_group"
-------------------------------------------------
  Default (fallback) Group:
cn=default_user_group,cn=groups,cn=accounts,dc=example,dc=com
```

**注意**

要删除当前的默认自动成员组，请输入 ipa automember-default-group-remove 命令。

验证步骤

-

要验证组是否已正确设置，请输入 ipa automember-default-group-show 命令。命令显示当前的默认的自动成员组。例如：

```
$ ipa automember-default-group-show
Grouping Type: group
  Default (fallback) Group:
cn=default_user_group,cn=groups,cn=accounts,dc=example,dc=com
```

# 第 18 章 使用 IDM WEB UI 自动化组成员资格

使用自动化组成员资格，使您可以根据其属性自动将用户和主机分配给组。例如，您可以：

- 根据员工的经理、位置或任何其他属性，将员工的用户条目划分为组。
- 根据主机的类、位置或任何其他属性来划分主机。
- 将所有用户或全部主机添加到单个全局组。

本章涵盖了以下主题：

- 使用 IdM Web UI 配置默认的用户组
- 使用 IdM Web UI 配置默认的主机组

## 18.1. 自动化组成员资格的好处

对用户使用自动成员资格，允许您：

- 减少手动管理组成员资格的开销

  您不再需要手动将每个用户和主机分配到组中。

- 提高用户和主机管理的一致性

  用户和主机根据严格定义的和自动评估的标准被分配到组。

- 简化基于组的设置的管理

  为组定义各种设置，然后应用到各个组成员，如 sudo 规则、自动挂载或访问控制。将用户和主机添加到组中会自动使管理这些设置变得更加简单。

## 18.2. 自动成员规则

在配置自动化组成员资格时，管理员定义自动成员规则。自动成员规则应用到特定的用户或主机目标组。它不能一次应用到多个组。

创建规则后，管理员会为其添加条件。它们指定将哪些用户或主机包含在目标组中，或从目标组中排除：

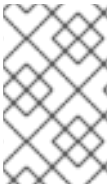
- 包含的条件

  当用户或主机条目满足包含的条件时，它将包含在目标组中。

- 排他条件

  当用户或主机条目满足排他条件时，它不会包含在目标组中。

条件被指定为 Perl 兼容的正则表达式(PCRE)格式的正则表达式。有关 PCRE 的更多信息，请参阅 pcresyntax (3) 手册页。



注意

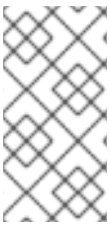
IdM 在包含条件之前评估排他条件。在发生冲突时，排他条件优先于包含条件。

自动成员规则适用于将来创建的每个条目。这些条目将自动添加到指定的目标组中。如果一个条目满足多个自动成员规则中指定的条件，它将被添加到所有对应的组中。

现有条目 不会受到 新规则的影响。如果要更改现有条目，请参阅 使用 IdM Web UI 将自动成员规则应用到现有条目。

### 18.3. 使用 IDM WEB UI 添加自动成员规则

按照以下流程，使用 IdM Web UI 添加自动成员规则。有关自动成员规则的信息，请参考 自动成员规则。



注意

现有条目 不会受到 新规则的影响。如果要更改现有条目，请参阅 使用 IdM Web UI 将自动成员规则应用到现有条目。

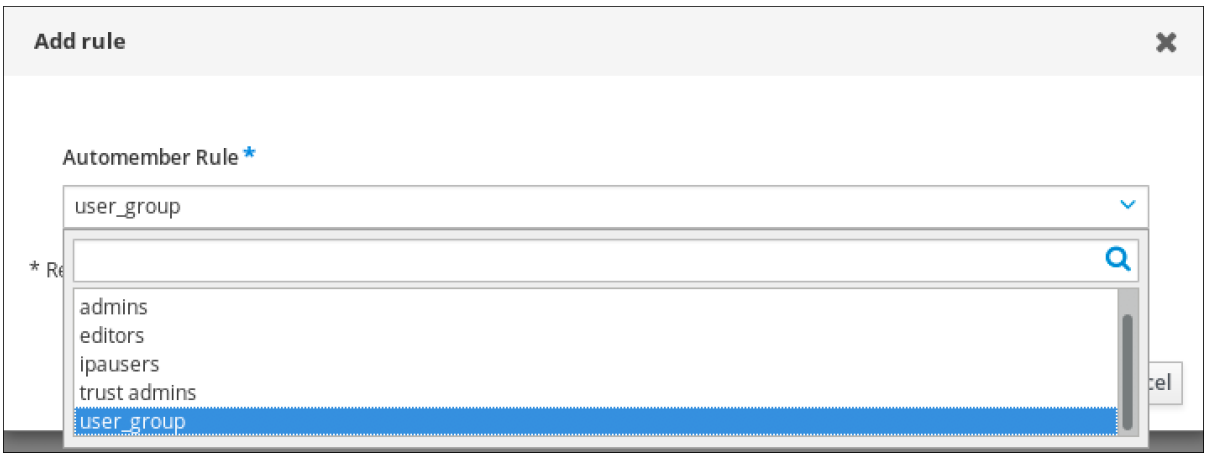
先决条件

- 已登陆到 IdM Web UI。
- 您必须是 admins 组的成员。

- 新规则的目标组在 IdM 中存在。

**流程**

1. 点击 Identity → Automember，然后选择 User group rules 或 Host group rules。
2. 点击 Add。
3. 在 Automember rule 字段中，选择规则要应用的组。这是目标组名称。




4. 单击 Add 确认。
5. 可选： 您可以使用在 使用 IdM Web UI 向自动成员规则中添加条件 中所述的步骤，向新规则添加条件。

## 18.4. 使用 IDM WEB UI 向自动成员规则中添加条件

配置自动成员规则后，您可以使用 IdM Web UI 向该自动成员规则添加条件。有关自动成员规则的信息，请参考 自动成员规则。

**先决条件**

- 已登陆到 IdM Web UI。
- 您必须是 admins 组的成员。

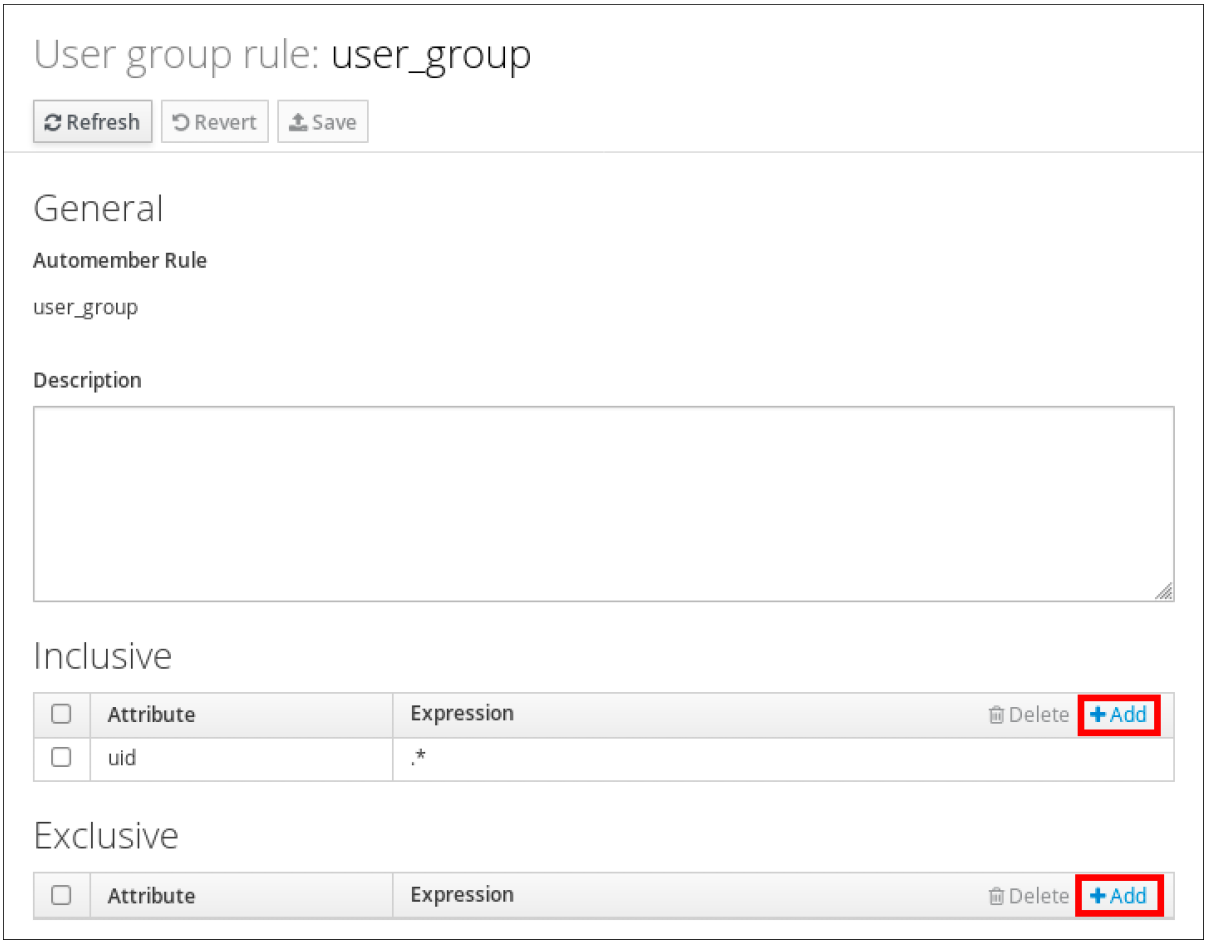
- 目标规则在 IdM 中存在。

**流程**

1. 点击 **Identity → Automember**，然后选择 **User group rules** 或 **Host group rules**。

2. 点击您要向其添加条件的规则。

3. 在 **Inclusive** 或 **Exclusive** 部分中，点击 **Add**。

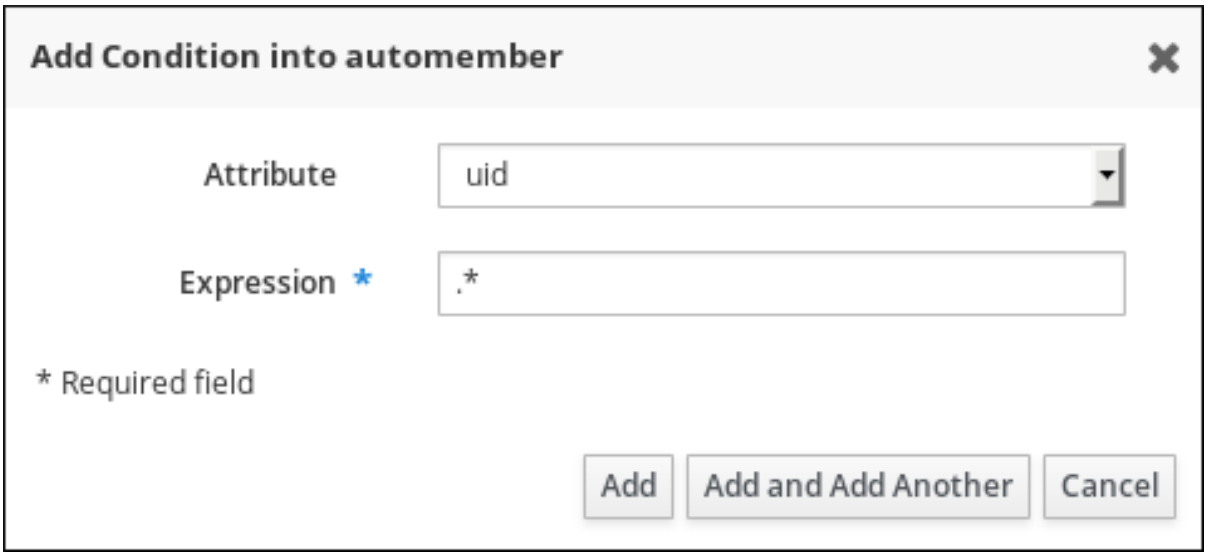



4. 在 **Attribute** 字段中，选择需要的属性，如 ***uid***。

5. 在 **Expression** 字段中，定义正则表达式。

6. 点击 **Add**。

例如，以下条件以用户 ID(uid)属性中带有任何值(.*)的所有用户为目标。




## 18.5. 使用 IDM WEB UI 查看现有的自动成员规则和条件

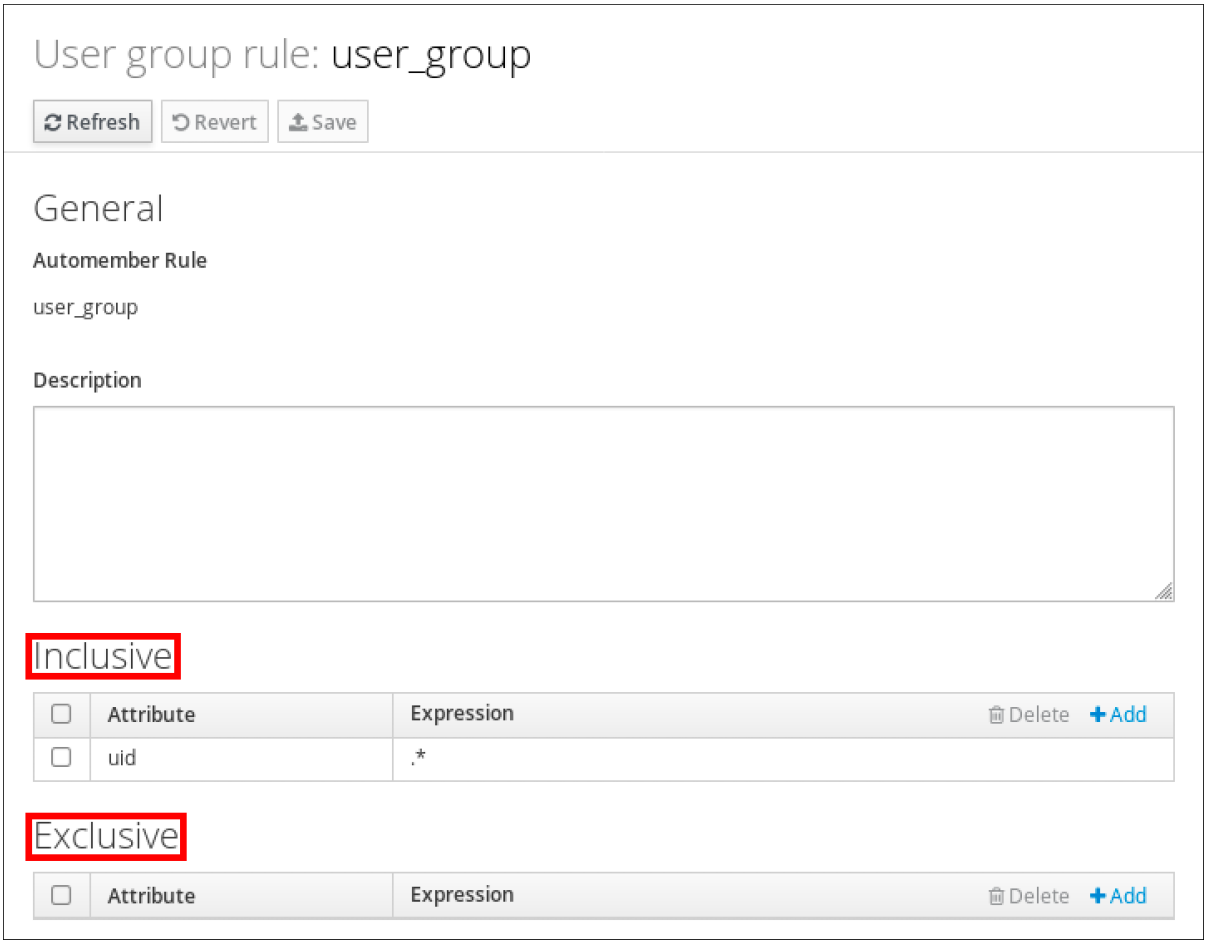
按照以下流程，使用 IdM Web UI 查看现有的自动成员规则和条件。

### 先决条件

- 已登陆到 IdM Web UI。
- 您必须是 admins 组的成员。

### 流程

1. 点击 Identity → Automember，然后选择 User group rules 或 Host group rules 来查看对应的自动成员规则。
2. 可选：点击规则，来查看 Inclusive 或 Exclusive 部分中规则的条件。




### 18.6. 使用 IDM WEB UI 删除自动成员规则

按照以下流程，使用 IdM Web UI 删除自动成员规则。

删除自动成员规则也会删除与规则相关的所有条件。要只从规则中删除特定条件，请参阅 使用 IdM Web UI 从自动成员规则中删除条件。

**先决条件**

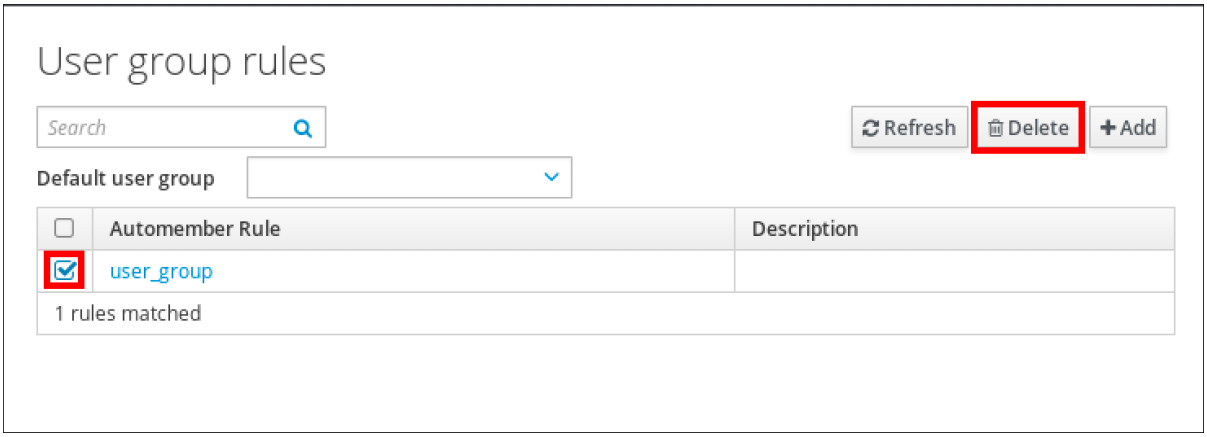
- 已登陆到 IdM Web UI。
- 您必须是 admins 组的成员。

**流程**

1. 点击 Identity → Automember，然后选择 User group rules 或 Host group rules 来查看对应的自动成员规则。

2. 选中您要删除的规则旁边的复选框。

3. 单击 **Delete**。




4. 单击 **Delete** 确认。

## 18.7. 使用 IDM WEB UI 从自动成员规则中删除条件

按照以下流程，使用 IdM Web UI 从自动成员规则中删除特定条件。

### 先决条件

- 已登陆到 IdM Web UI。
- 您必须是 admins 组的成员。

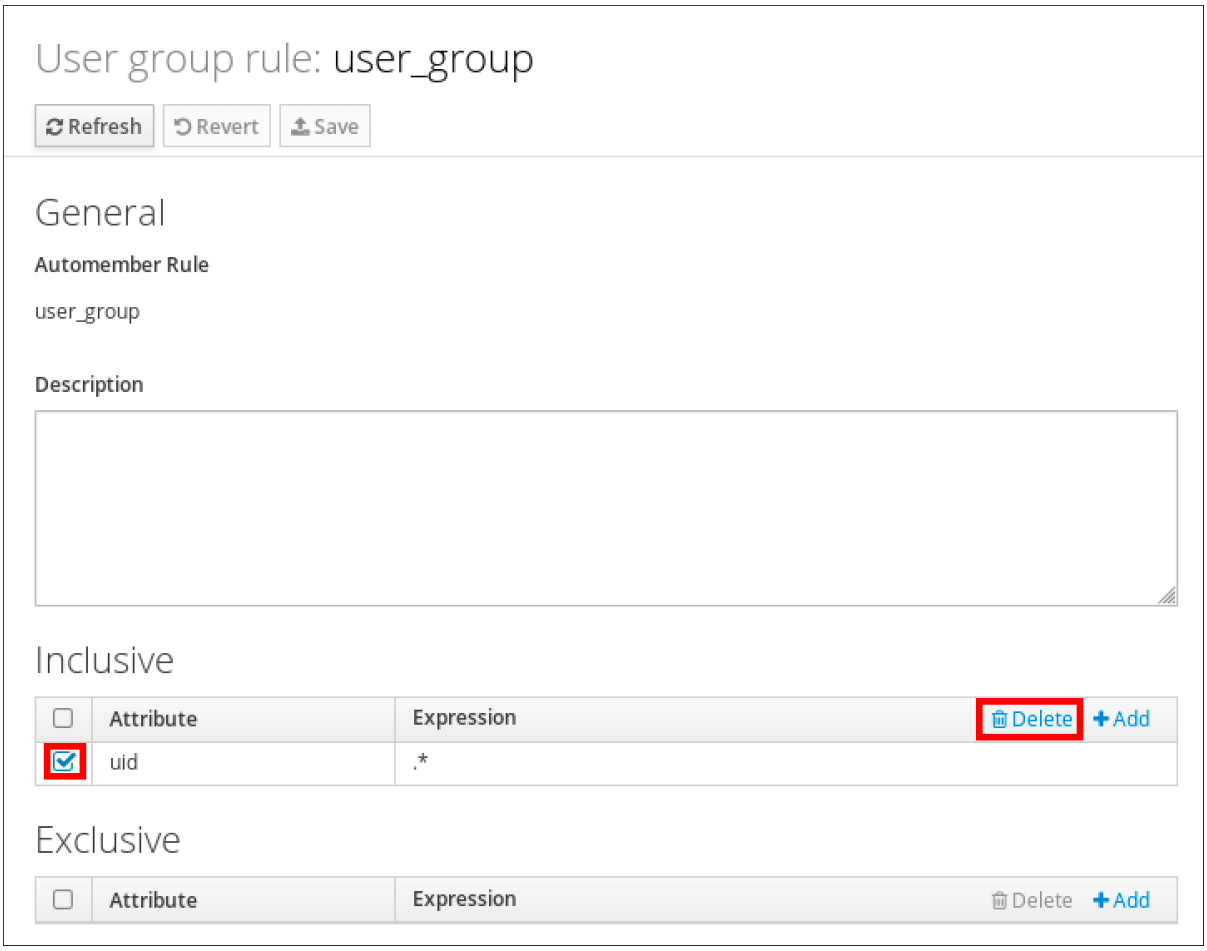
### 流程

1. 点击 **Identity** → **Automember**，然后选择 **User group rules** 或 **Host group rules** 来查看对应的自动成员规则。

2. 点击规则，来查看 **Inclusive** 或 **Exclusive** 部分中规则的条件。

3. 选中您要删除的条件旁边的复选框。

4. 单击 **Delete**。




5. 单击 **Delete** 确认。

### 18.8. 使用 IDM WEB UI 将自动成员规则应用到现有条目

自动成员规则在规则添加后，自动应用到所创建的用户和主机条目。它们不会追溯到在规则添加之前存在的条目。

要将自动成员规则应用到之前添加的条目，您必须手动重建自动成员资格。重建自动成员资格会重新评估所有现有的自动成员规则，并将其应用到所有用户或主机条目或特定的条目。

**注意**

重建自动成员资格 不会 从组中删除用户或主机条目，即使条目不再与组的包含条件匹配。要手动删除它们，请参阅 使用 IdM Web UI 从用户组中删除成员 或 在 IdM Web UI 中删除主机组成员。

### 18.8.1. 为所有用户或主机重建自动成员资格

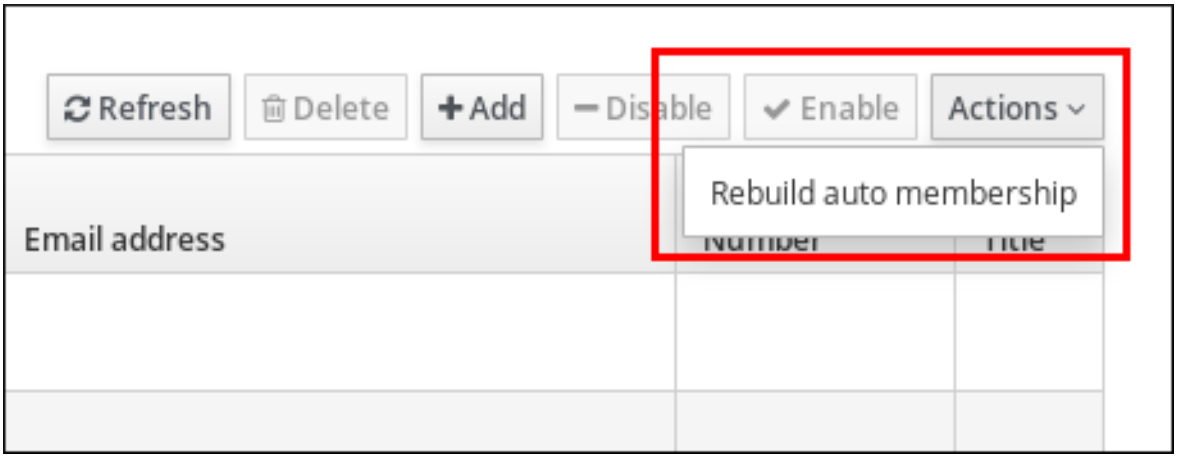
**按照以下流程，为所有用户或主机条目重建自动成员资格。**

**先决条件**

- **已登陆到 IdM Web UI。**
- **您必须是 admins 组的成员。**

**流程**

1. 选择 **Identity → Users 或 Hosts。**
2. 单击 **Actions → Rebuild auto membership。**




### 18.8.2. 只为单个用户或主机重建自动成员资格

**按照以下流程，为特定用户或主机条目重建自动成员资格。**

**先决条件**

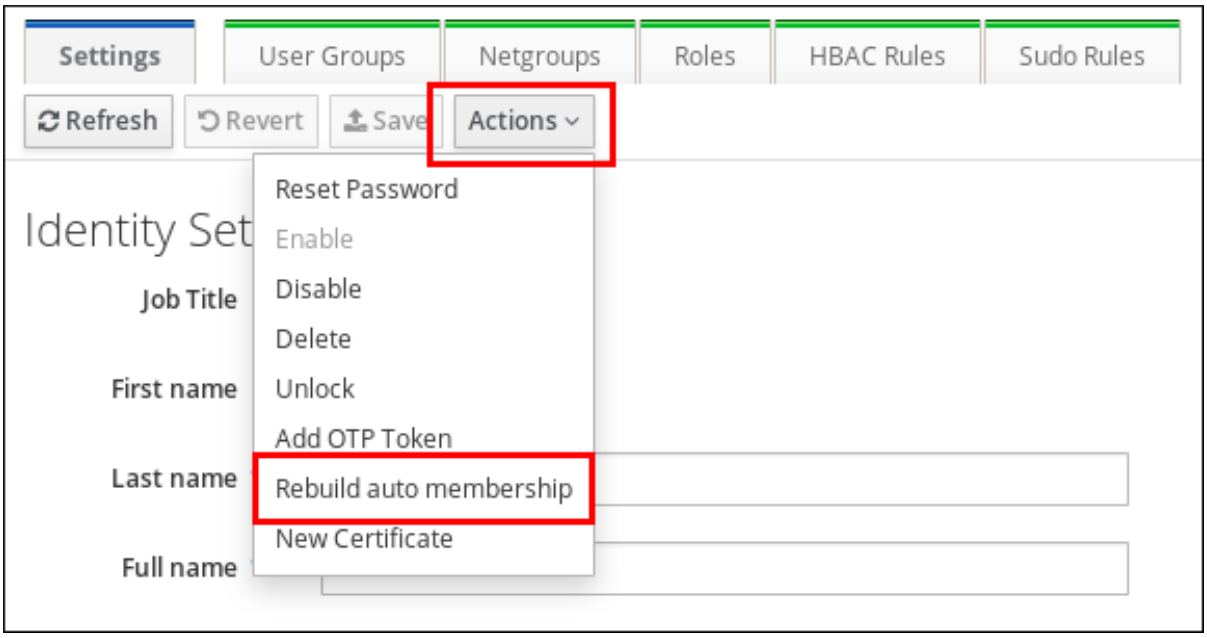
- **已登陆到 IdM Web UI。**
- **您必须是 admins 组的成员。**

**流程**

1. 选择 **Identity → Users** 或 **Hosts**。

2. 单击所需的用户或主机名。

3. 单击 **Actions → Rebuild auto membership**。




## 18.9. 使用 IDM WEB UI 配置默认的用户组

**当您配置默认用户组时，不与任何自动成员规则匹配的新用户条目将自动添加到此默认组中。**

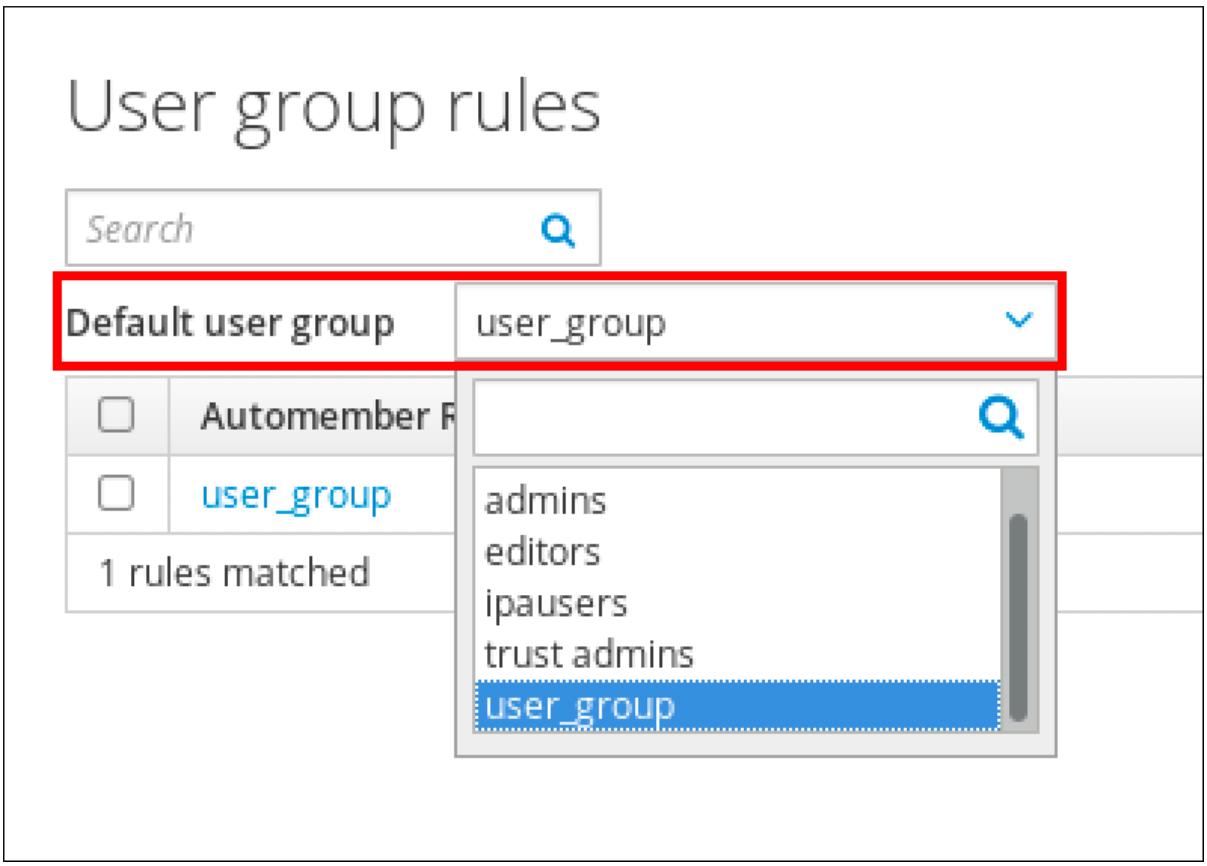
**先决条件**

- **已登陆到 IdM Web UI。**
- **您必须是 admins 组的成员。**
- **您要设置为默认的目标用户组在 IdM 中存在。**

**流程**

1. 点击 **Identity → Automember**，然后选择 **User group rules**。

2. 在 **Default user group** 字段中，选择您要设置为默认用户组的组。




## 18.10. 使用 IDM WEB UI 配置默认的主机组

配置默认主机组时，不与任何自动成员规则匹配的新主机条目将自动添加到此默认组中。

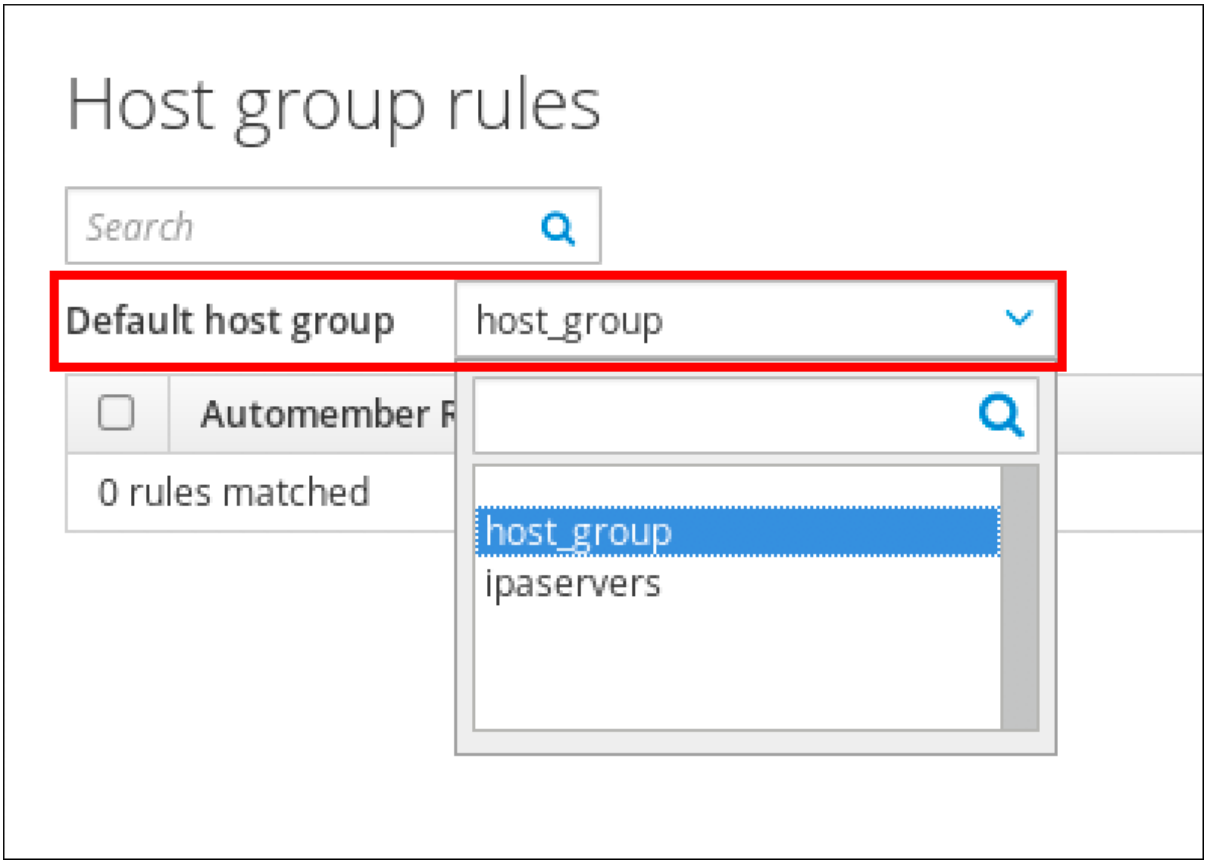
**先决条件**

- 已登陆到 IdM Web UI。
- 您必须是 **admins** 组的成员。
- 您要设置为默认的目标主机组在 IdM 中存在。

**流程**

1. **点击 Identity → Automember，然后选择 Host group rules。**

2. **在 Default host group 字段中，选择您要设置为默认主机组的组。**



# 第 19 章 使用 ANSIBLE 在 IDM 中自动化组成员资格

通过自动化组成员资格，您可以根据其属性自动分配用户、主机用户组和主机组。例如，您可以：

- 根据员工的经理、地点、职位或任何其他属性将员工的用户条目分成不同的组。您可以通过在命令行中输入 ipa user-add --help 来列出所有属性。
- 根据它们的类、位置或任何其他属性，将主机分成不同的组。您可以通过在命令行中输入 ipa host-add --help 来列出所有属性。
- 将所有用户或全部主机添加到单个全局组。

您可以使用 Red Hat Ansible Engine 来自动管理身份管理(IdM)中的自动化组成员资格。

本节涵盖了以下主题：

- 准备 Ansible 控制节点来管理 IdM
- 使用 Ansible 确保 IdM 用户组的自动成员规则存在
- 使用 Ansible 确保 IdM 用户组自动成员规则中存在条件
- 使用 Ansible 确保 IdM 用户组自动成员规则中的条件不存在
- 使用 Ansible 确保 IdM 组的自动成员规则不存在
- 使用 Ansible 确保 IdM 主机组自动成员规则中存在条件

## 19.1. 准备 ANSIBLE 控制节点来管理 IDM

作为管理身份管理 (IdM) 的系统管理员，在使用 Red Hat Ansible Engine 时，最好执行以下操作：

- 在您的主目录中，创建专用于 Ansible playbook 的子目录，如 ~/MyPlaybooks。
- 将示例 Ansible playbook 从 /usr/share/doc/ansible-freeipa/* 和 /usr/share/doc/rhel-system-roles/* 目录以及它们的子目录复制到 ~/MyPlaybooks 目录中并进行调整。
- 将清单文件包含在 ~/MyPlaybooks 目录中。

按照这种做法，您可以在一个地方找到所有 playbook，您可以在不调用 root 特权的情况下运行 playbook。

注意

您只需要受管主机上的 root 权限来执行 ipaserver、ipareplica、ipaclient、ipabackup、ipasmartcard_server 和 ipasmartcard_client ansible-freeipa 角色。这些角色需要具有目录和 dnf 软件包管理器的特权访问权限。

按照以下流程创建 ~/MyPlaybooks 目录并进行配置，以便您可以使用它来存储和运行 Ansible playbook。

先决条件

- 您已在受管节点上安装了 IdM 服务器 *server.idm.example.com* 和 *replica.idm.example.com*。
- 您已配置了 DNS 和网络，以便您可以直接从控制节点登录到受管节点 *server.idm.example.com* 和 *replica.idm.example.com*。
- 您需要知道 IdM admin 密码。

流程

1. 在主目录中为您的 Ansible 配置和 playbook 创建目录：

```
$ mkdir ~/MyPlaybooks/
```

2. 进入 ~/MyPlaybooks/ 目录：

```
$ cd ~/MyPlaybooks
```

3. 使用以下内容创建 ~/MyPlaybooks/ansible.cfg 文件：

```
[defaults]
inventory = /home/your_username/MyPlaybooks/inventory

[privilege_escalation]
become=True
```

4. 使用以下内容创建 ~/MyPlaybooks/inventory 文件：

```
[ipaserver]
server.idm.example.com

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com

[ipacluster:children]
ipaserver
ipareplicas

[ipacluster:vars]
ipaadmin_password=SomeADMINpassword

[ipaclients]
ipaclient1.example.com
ipaclient2.example.com

[ipaclients:vars]
ipaadmin_password=SomeADMINpassword
```

此配置定义了两个主机组，即 eu 和 us，用于这些位置中的主机。此外，此配置定义了 ipaserver 主机组，它包含来自 eu 和 us 组的所有主机。

5. [可选] 创建 SSH 公钥和私钥。要在测试环境中简化访问，请不要在私钥中设置密码：

```
$ ssh-keygen
```

6. 将 SSH 公钥复制到每个受管节点上的 IdM admin 帐户：

```
$ ssh-copy-id admin@server.idm.example.com
$ ssh-copy-id admin@replica.idm.example.com
```

输入这些命令时，您必须输入 IdM admin 密码。

其它资源

- 使用 Ansible playbook 安装身份管理服务器。
- 如何构建清单。

## 19.2. 使用 ANSIBLE 确保 IDM 用户组的自动成员规则存在

以下流程描述了如何使用 Ansible playbook 确保身份管理(IdM)组的 自动成员 规则存在。在示例中，确保 testing_group 用户组的 自动成员 规则存在。

先决条件

- 您需要知道 IdM admin 密码。
- IdM 中存在 testing_group 用户组。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/automember/ 目录中的 automember-group-present.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-group-present.yml automember-group-present-copy.yml
```

3. 打开 automember-group-present-copy.yml 文件进行编辑。

4. 通过在 ipaautomember 任务部分中设置以下变量来调整该文件：

   - 将 ipaadmin_password 变量设置为 IdM admin 的密码。
   - 将 name 变量设为 testing_group。
   - 将 automember_type 变量设为 group。
   - 确保 state 变量设置为 present。

   这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Automember group present example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure group automember rule admins is present
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: testing_group
      automember_type: group
      state: present
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-group-present-copy.yml
```

**其它资源**

- 查看 自动化组成员资格的好处 和 自动成员规则 。
- 请参阅 使用 Ansible 来确保 IdM 用户组 自动成员规则中存在条件。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-automember.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/automember 目录。

## 19.3. 使用 ANSIBLE 确保指定的条件在 IDM 用户组自动成员规则中存在

以下流程描述了如何使用 Ansible playbook 来确保指定的条件在身份管理(IdM)组的 自动成员 规则中存在。在示例中，确保 testing_group 组的 自动成员 规则中存在与 UID 相关的条件。通过指定 .* 条件，您可以确保所有将来的 IdM 用户都自动成为 testing_group 的成员。

**先决条件**

- 您需要知道 IdM admin 密码。
- testing_group 用户组和自动成员用户组规则在 IdM 中存在。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/automember/ 目录中的 automember-hostgroup-rule-present.yml Ansible playbook 文件，并将它命名为 automember-usergroup-rule-present.yml ：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-present.yml automember-usergroup-rule-present.yml
```

3. 打开 automember-usergroup-rule-present.yml 文件进行编辑。

4. 通过修改以下参数来调整文件：

- 重命名 playbook 以便对应于您的用例，例如：自动成员用户组规则成员存在。
- 重命名任务以便对应于您的用例，例如：确保用户组的自动成员条件存在。
- 在 ipaautomember 任务部分中设置以下变量：
  - 将 ipaadmin_password 变量设置为 IdM admin 的密码。
  - 将 name 变量设为 testing_group。
  - 将 automember_type 变量设为 group。
  - 确保 state 变量设置为 present。
  - 确保 action 变量设为 member。
  - 将 inclusive key 变量设为 UID。
  - 将 inclusive expression 变量设为 .*

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Automember user group rule member present
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure an automember condition for a user group is present
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
```

```
    name: testing_group
    automember_type: group
    state: present
    action: member
    inclusive:
      - key: UID
        expression: .*
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-usergroup-rule-present.yml
```

验证步骤

1. 以 IdM 管理员身份登录。

```
$ kinit admin
```

2. 例如，添加用户：

```
$ ipa user-add user101 --first user --last 101
----------------------
Added user "user101"
----------------------
  User login: user101
  First name: user
  Last name: 101
  ...
  Member of groups: ipausers, testing_group
  ...
```

其它资源

- 请参阅 使用 IdM CLI 将自动成员规则应用到现有条目。
- 查看 自动化组成员资格的好处 和 自动成员规则 。

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-automember.md 文件。

- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/automember 目录。

## 19.4. 使用 ANSIBLE 确保条件在 IDM 用户组自动成员规则中不存在

以下流程描述了如何使用 Ansible playbook 确保条件在身份管理(IdM)组的 自动成员 规则中不存在。在示例中，条件在 自动成员 规则中不存在确保了应包含指定 首字母 为 dp 的用户。将自动成员规则应用到 testing_group 组。通过应用条件，您可以确保将来首字母为 dp 的用户不会成为 testing_group 的成员。

### 先决条件

- 您需要知道 IdM admin 密码。

- testing_group 用户组和自动成员用户组规则在 IdM 中存在。

- 您已配置了 Ansible 控制节点以满足以下要求：

  - 您使用 Ansible 版本 2.14 或更高版本。

  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。

  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。

  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

### 流程

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/automember/ 目录中的 automember-hostgroup-rule-absent.yml Ansible playbook 文件，并将其命名为 automember-usergroup-rule-absent.yml ：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-absent.yml automember-usergroup-rule-absent.yml
```

3. 打开 automember-usergroup-rule-absent.yml 文件进行编辑。

4. 通过修改以下参数来调整文件：

- 重命名 playbook 以对应于您的用例，例如：自动成员用户组规则成员不存在。
- 重命名任务以对应于您的用例，例如：确保用户组的自动成员条件不存在。
- 在 ipaautomember 任务部分中设置以下变量：
  - 将 ipaadmin_password 变量设置为 IdM admin 的密码。
  - 将 name 变量设为 testing_group。
  - 将 automember_type 变量设为 group。
  - 确保 state 变量设置为 absent。
  - 确保 action 变量设为 member。

- 将 inclusive key 变量设为 initials。
- 将 inclusive expression 变量设为 dp。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Automember user group rule member absent
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure an automember condition for a user group is absent
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: testing_group
      automember_type: group
      state: absent
      action: member
      inclusive:
        - key: initials
          expression: dp
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-usergroup-rule-absent.yml
```

验证步骤

1. 以 IdM 管理员身份登录。

```
$ kinit admin
```

2. 查看自动成员组：

```
$ ipa automember-show --type=group testing_group
 Automember Rule: testing_group
```

输出中没有 Inclusive Regex: initials=dp 条目确认 testing_group 自动成员规则不包含指定的条件。

其它资源

- 请参阅 使用 IdM CLI 将自动成员规则应用到现有条目。
- 查看 自动化组成员资格的好处 和 自动成员规则 。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-automember.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/automember 目录。

## 19.5. 使用 ANSIBLE 确保 IDM 用户组的自动成员规则不存在

以下流程描述了如何使用 Ansible playbook 确保身份管理(IdM)组的 自动成员 规则不存在。在示例中，确保 testing_group 组的 automember 规则不存在。

注意

删除自动成员规则也会删除与规则相关的所有条件。要从规则中只删除特定的条件，请参阅 使用 Ansible 确保条件在 IdM 用户组自动成员规则中不存在。

先决条件

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。

- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/automember/ 目录中的 automember-group-absent.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-group-absent.yml automember-group-absent-copy.yml
```

3. 打开 automember-group-absent-copy.yml 文件进行编辑。

4. 通过在 ipaautomember 任务部分中设置以下变量来调整该文件：

- 将 ipaadmin_password 变量设置为 IdM admin 的密码。
- 将 name 变量设为 testing_group。
- 将 automember_type 变量设为 group。
- 确保 state 变量设置为 absent。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Automember group absent example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure group automember rule admins is absent
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: testing_group
      automember_type: group
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-group-absent.yml
```

**其它资源**

- 查看 自动化组成员资格的好处 和 自动成员规则 。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-automember.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/automember 目录。

## 19.6. 使用 ANSIBLE 确保 IDM 主机组自动成员规则中存在条件

按照以下流程，使用 Ansible 确保条件在 IdM 主机组自动成员规则中存在。示例描述了如何确保 FQDN 为 .*.idm.example.com 的主机是 primary_dns_domain_hosts 主机组的成员，以及 FQDN 为 .*.example.org 的主机不是 primary_dns_domain_hosts 主机组的成员。

**先决条件**

- 您需要知道 IdM admin 密码。
- IdM 中存在 primary_dns_domain_hosts 主机组和自动成员主机组规则。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/automember/ 目录中的 automember-hostgroup-rule-present.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-present.yml automember-hostgroup-rule-present-copy.yml
```

3. 打开 automember-hostgroup-rule-present-copy.yml 文件进行编辑。

4. 通过在 ipaautomember 任务部分中设置以下变量来调整该文件：

- 将 ipaadmin_password 变量设置为 IdM admin 的密码。
- 将 name 变量设为 primary_dns_domain_hosts。
- 将 automember_type 变量设为 hostgroup。
- 确保 state 变量设置为 present。
- 确保 action 变量设为 member。
- 确保 inclusive key 变量设为 fqdn。
- 将对应的 inclusive expression 变量设为 .*.idm.example.com。
- 将 exclusive key 变量设为 fqdn。
- 将对应的 exclusive expression 变量设为 .*.example.org。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Automember user group rule member present
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure an automember condition for a user group is present
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: primary_dns_domain_hosts
      automember_type: hostgroup
      state: present
      action: member
```

```
    inclusive:
      - key: fqdn
        expression: .*.idm.example.com
    exclusive:
      - key: fqdn
        expression: .*.example.org
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-hostgroup-rule-present-copy.yml
```

**其它资源**

- 请参阅 使用 IdM CLI 将自动成员规则应用到现有条目。
- 查看 自动化组成员资格的好处 和 自动成员规则 。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-automember.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/automember 目录。

## 19.7. 其它资源

- 使用 Ansible playbook 管理用户帐户
- 使用 Ansible playbook 管理主机
- 使用 Ansible playbook 管理用户组
- 使用 IdM CLI 管理主机组

# 第 20 章 IDM 中的访问控制

访问控制定义了授予用户对其他用户或对象（如主机或服务）执行操作的权利或权限。身份管理(IdM)提供了多个访问控制区，以明确授予了哪些类型的访问权限，以及权限被授予给了谁。因此，IdM 会区分了对域中资源的访问控制和对 IdM 配置本身的访问控制。

本章概述了 IdM 用户对域内资源以及对 IdM 配置本身的不同的内部访问控制机制。

## 20.1. IDM 中的访问控制指令

身份管理(IdM)访问控制结构是基于 389 目录服务器访问控制的。通过使用访问控制指令(ACI)，您可以授予或拒绝特定的 IdM 用户对其他条目的访问。所有条目（包括 IdM 用户）都存储在 LDAP 中。

ACI 有三个部分：

行动者

被授予权限可以做某事的实体。在 LDAP 访问控制模型中，您可以指定 ACI 规则只有在用户使用其可区分的名称(DN)绑定到目录时才应用。此类规格称为 *绑定规则* ：它会定义用户是谁，并可以选择要求对绑定尝试的其他限制，例如将尝试限制在一天的某段时间或某台机器上。

目标

允许行动者对其执行操作的条目。

操作类型

确定行动者可以执行哪种操作。最常见的操作有 add、delete、write、read 和 search。在 IdM 中，非管理员用户的读和搜索权限是有限制的，IdM Web UI 中的限制比 IdM CLI 中的限制更多。

当尝试 LDAP 操作时，会出现以下情况：

1. IdM 客户端将用户凭证发送到 IdM 服务器，作为绑定 操作的一部分。
2. IdM 服务器 DS 检查用户凭证。
3. IdM 服务器 DS 检查用户帐户，以查看用户是否有执行所请求的操作的权限。

## 20.2. IDM 中的访问控制方法

身份管理(IdM)将访问控制方法分为以下类别：

### 自助服务规则

定义用户对其自己的个人条目可以执行哪些操作。此访问控制类型仅允许对用户条目中的特定属性具有写权限。用户可以更新特定属性的值，但不能添加或删除这些属性。

### 委派规则

通过使用委派规则，您可以允许特定的用户组对另一个用户组中特定的用户属性执行写（也就是编辑）操作。与自助服务规则类似，这种形式的访问控制规则被限制为编辑特定属性的值。它不能授予添加或删除整个条目或控制未指定属性的权限。

### 基于角色的访问控制

创建特殊的访问控制组，然后对 IdM 域中的所有实体类型授予更大的权力。可以授予角色编辑、添加和删除的权限，即可以授予它们对整个条目的完整控制，而不仅仅是对所选择的属性。

默认情况下，IdM 中已经提供了某些角色，如 Enrollment Administrator、IT Security Specialist 和 IT Specialist。您可以创建额外的角色来管理任何类型的条目，如主机、自动挂载配置、网络组、DNS 设置和 IdM 配置。

### 其它资源

- 使用 Ansible playbook 管理 IdM 中的自助服务规则
- 委派权限到用户组，以使用 Ansible playbook 管理用户
- 在 IdM 中使用 Ansible playbook 管理基于角色的访问控制

# 第 21 章 使用 CLI 管理 IDM 中的自助服务规则

了解身份管理(IdM)中的自助服务规则，以及如何在命令行界面(CLI)中创建和编辑自助服务访问规则。

## 21.1. IDM 中的自助服务访问控制

自助服务访问控制规则定义 Identity Management (IdM) 实体可以在其 IdM 目录服务器条目上执行的操作：例如，IdM 用户能够更新自己的密码。

这种控制方法允许经过身份验证的 IdM 实体编辑其 LDAP 条目中的特定属性，但不允许对整个条目的 add 或 delete 操作。

**警告**

使用自助服务访问控制规则时要小心：不当配置访问控制规则可能会意外地提升实体的特权。

## 21.2. 使用 CLI 创建自助服务规则

按照以下流程，使用命令行界面(CLI)在 IdM 中创建自助服务访问规则。

**先决条件**

- 管理 IdM 或 用户管理员 角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

- 要添加自助服务规则，请使用 ipa selfservice-add 命令，并指定以下两个选项：

--permissions

设置访问控制指令(ACI)授予的 读 和 写 权限。

--attrs

设置此 ACI 授予权限的属性的完整列表。

例如，要创建一个自助服务规则，允许用户修改其自己的名称详情：

```
$ ipa selfservice-add "Users can manage their own name details" --permissions=write --
attrs=givenname --attrs=displayname --attrs=title --attrs=initials
--------------------------------------------------------
Added selfservice "Users can manage their own name details"
--------------------------------------------------------
  Self-service name: Users can manage their own name details
  Permissions: write
  Attributes: givenname, displayname, title, initials
```

### 21.3. 使用 CLI 编辑自助服务规则

按照以下流程，使用命令行界面(CLI)编辑 IdM 中的自助服务访问规则。

**先决条件**

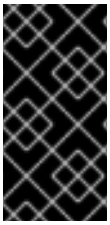
- 管理 IdM 或 用户管理员 角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. *可选* ：使用 ipa selfservice-find 命令显示现有的自助服务规则。
2. *可选* ：使用 ipa selfservice-show 命令显示您要修改的自助服务规则的详情。
3. 使用 ipa selfservice-mod 命令来编辑自助服务规则。

例如：

```
$ ipa selfservice-mod "Users can manage their own name details" --attrs=givenname --
attrs=displayname --attrs=title --attrs=initials --attrs=surname
-------------------------------------------------------------
Modified selfservice "Users can manage their own name details"
-------------------------------------------------------------
Self-service name: Users can manage their own name details
Permissions: write
Attributes: givenname, displayname, title, initials
```



**重要**

使用 ipa selfservice-mod 命令覆盖之前定义的权限和属性，因此始终包含现有权限和属性的完整列表，以及您要定义的任何新的权限和属性。

验证步骤

- 使用 ipa selfservice-show 命令显示您编辑的自助服务规则。

```
$ ipa selfservice-show "Users can manage their own name details"
-------------------------------------------------------------
Self-service name: Users can manage their own name details
Permissions: write
Attributes: givenname, displayname, title, initials
```

### 21.4. 使用 CLI 删除自助服务规则

按照以下流程，使用命令行界面(CLI)删除 IdM 中的自助服务访问规则。

**先决条件**

- 管理 IdM 或 用户管理员 角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

- 使用 ipa selfservice-del 命令删除自助服务规则。

例如：

```
$ ipa selfservice-del "Users can manage their own name details"
-----------------------------------------------------------
Deleted selfservice "Users can manage their own name details"
-----------------------------------------------------------
```

验证步骤

- 使用 ipa selfservice-find 命令显示所有自助服务规则。您刚才删除的规则应该消失了。

# 第 22 章 使用 IDM WEB UI 管理自助服务规则

了解身份管理(IdM)中的自助服务规则，以及如何在 Web 界面(IdM Web UI)中创建和编辑自助服务访问规则。

## 22.1. IDM 中的自助服务访问控制

自助服务访问控制规则定义 Identity Management (IdM) 实体可以在其 IdM 目录服务器条目上执行的操作：例如，IdM 用户能够更新自己的密码。

这种控制方法允许经过身份验证的 IdM 实体编辑其 LDAP 条目中的特定属性，但不允许对整个条目的 add 或 delete 操作。

**警告**

使用自助服务访问控制规则时要小心：不当配置访问控制规则可能会意外地提升实体的特权。

## 22.2. 使用 IDM WEB UI 创建自助服务规则

按照以下流程，使用 Web 界面(IdM Web UI)在 IdM 中创建自助服务访问规则。

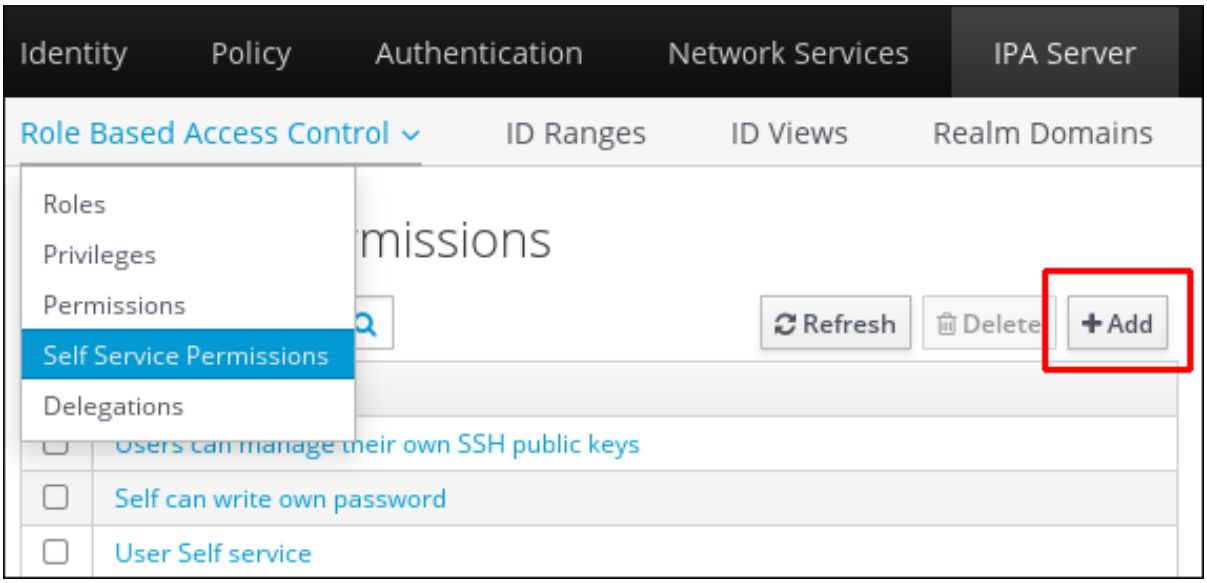
**先决条件**

- 管理 IdM 或 用户管理员 角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

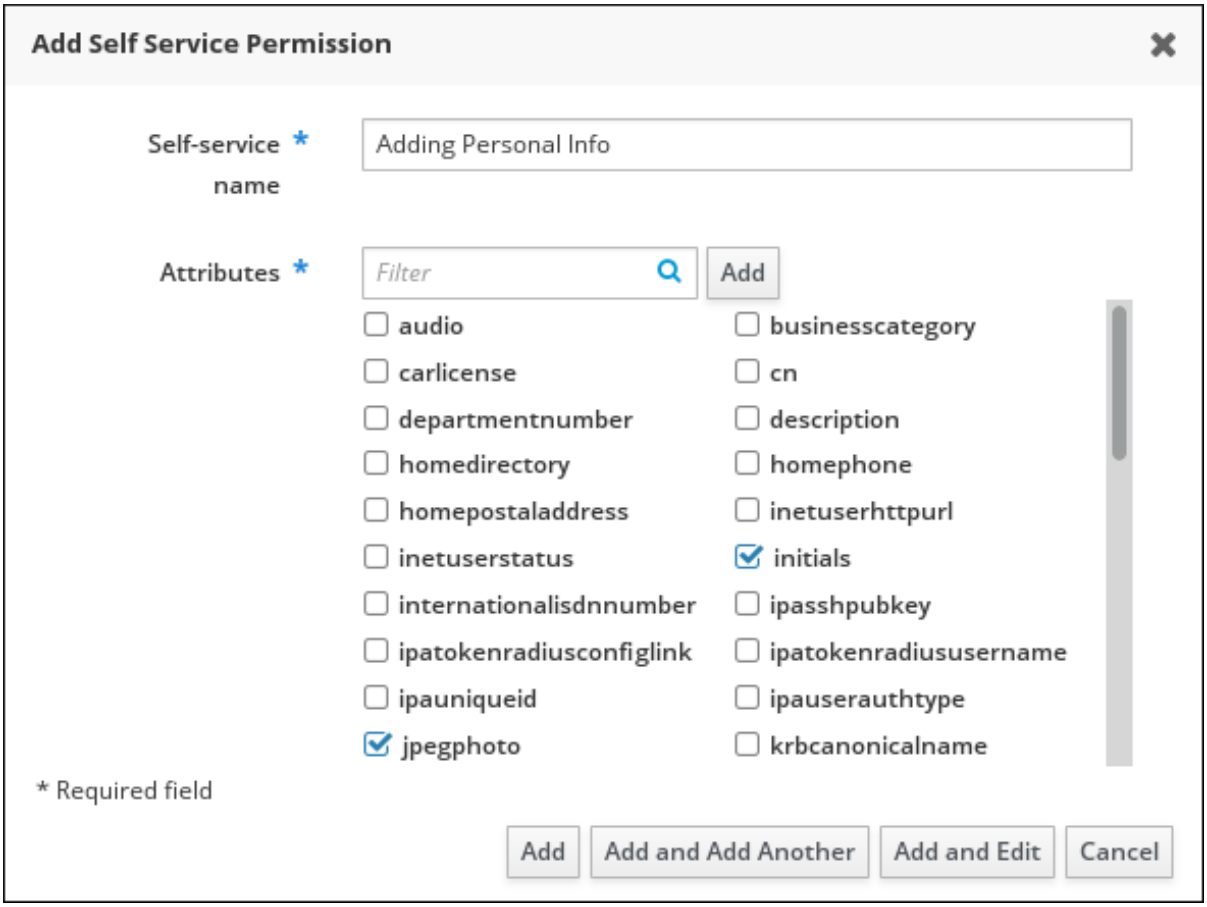
**流程**

1. 在 IPA Server 选项卡中，打开 Role-Based Access Control 子菜单，然后选择 Self Service Permissions。

2. **点自助服务访问规则列表右上角的 Add ：**




3. **此时将打开 Add Self Service Permission 窗口。在 Self-service name 字段中输入新自助服务规则的名称。允许空格：**



4. 选中您希望用户能够编辑的属性旁边的复选框。

5. *可选*：如果您要对其提供访问权限的属性没有列出，您可以为它添加一个列表：

   a. 点击 Add 按钮。

   b. 在以下 Add Custom Attribute 窗口的 Attribute 文本字段中输入属性名称。

   c. 单击 OK 按钮来添加该属性

   d. 验证是否已选中新属性

6. 单击表单底部的 Add 按钮，来保存新的自助服务规则。
或者，您可以通过单击 Add and Edit 按钮来保存并继续编辑自助服务规则，或者通过单击 Add and Add another 按钮来保存并添加其他规则。

## 22.3. 使用 IDM WEB UI 编辑自助服务规则

按照以下流程，使用 Web 界面(IdM Web UI)编辑 IdM 中的自助服务访问规则。

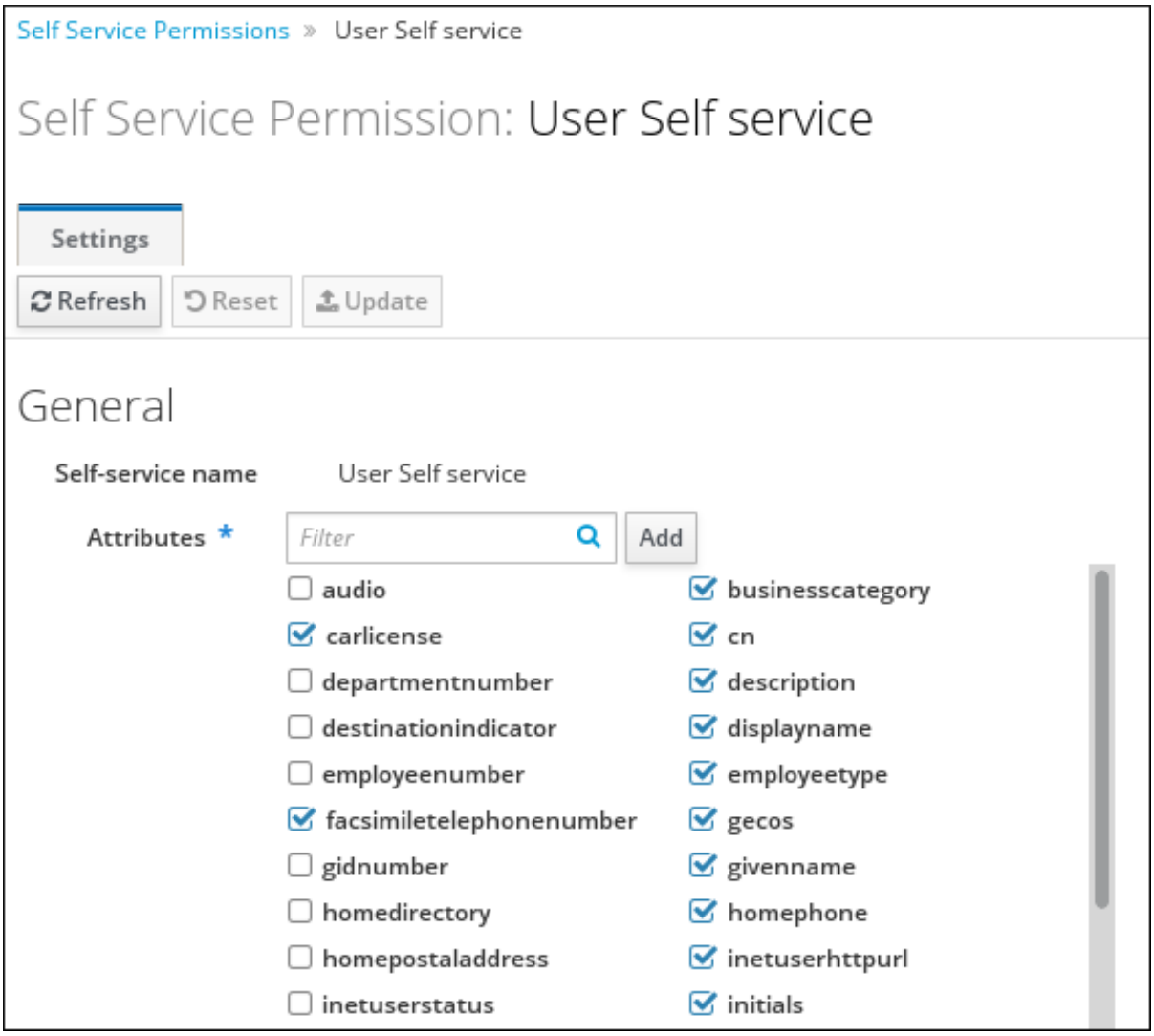
**先决条件**

- 管理 IdM 或 用户管理员 角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

**流程**

1. 在 IPA Server 选项卡中，打开 Role-Based Access Control 子菜单，然后选择 Self Service Permissions。

2. 单击您要修改的自助服务规则的名称。




3. 编辑页面只允许您编辑您要添加或删除自助服务规则的属性列表。选择或取消选择合适的复选框。

4. 单击 Save 按钮，将更改保存到自助服务规则。

## 22.4. 使用 IDM WEB UI 删除自助服务规则

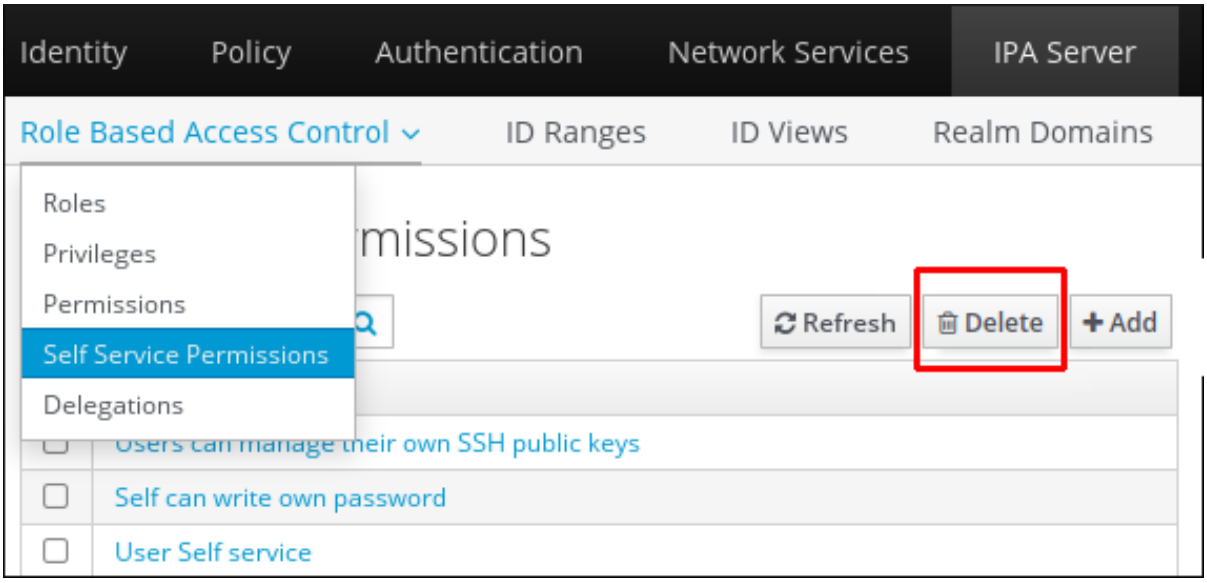
按照以下流程，使用 Web 界面(IdM Web UI)删除 IdM 中的自助服务访问规则。

**先决条件**

- 管理 IdM 或 用户管理员 角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

**流程**

1. 在 IPA Server 选项卡中，打开 Role-Based Access Control 子菜单，然后选择 Self Service Permissions。
2. 选中您要删除的规则旁边的复选框，然后单击列表右侧的 Delete 按钮。




3. 此时会打开一个对话框，单击 Delete 进行确认。

# 第 23 章 使用 ANSIBLE PLAYBOOK 管理 IDM 中的自助服务规则

本节介绍 Identity Management (IdM) 中的自助服务规则，并介绍如何使用 Ansible playbook 创建和编辑自助服务访问规则。自助服务访问控制规则允许 IdM 实体在其 IdM 目录服务器条目上执行指定操作。

- IdM 中的自助服务访问控制
- 使用 Ansible 确保存在自助服务规则
- 使用 Ansible 确保缺少自助服务规则
- 使用 Ansible 确保自助服务规则具有特定属性
- 使用 Ansible 确保自助服务规则没有特定属性

## 23.1. IDM 中的自助服务访问控制

自助服务访问控制规则定义 Identity Management (IdM) 实体可以在其 IdM 目录服务器条目上执行的操作：例如，IdM 用户能够更新自己的密码。

这种控制方法允许经过身份验证的 IdM 实体编辑其 LDAP 条目中的特定属性，但不允许对整个条目的 add 或 delete 操作。

**警告**

使用自助服务访问控制规则时要小心：不当配置访问控制规则可能会意外地提升实体的特权。

## 23.2. 使用 ANSIBLE 确保存在自助服务规则

以下流程描述了如何使用 Ansible playbook 定义自助服务规则并确保它们在身份管理 (IdM) 服务器上存在。在本例中，新的 Users can manage their own name details 规则会授予用户更改其 givenname、displayname、title 和 initials 属性的权限。例如，这允许他们更改其显示名称或缩写（如果想更改）。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/selfservice/ 目录中的 selfservice-present.yml 文件副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-present.yml selfservice-present-copy.yml
```

3. 打开 selfservice-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipaselfservice 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为新自助服务规则的名称。
- 将 permission 变量设置为以逗号分隔的权限列表，以授予：read 和 write。
- 将 attribute 变量设置为用户可以自己管理的属性列表：givenname、displayname、title 和 initials。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Self-service present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure self-service rule "Users can manage their own name details" is
present
    ipaselfservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "Users can manage their own name details"
      permission: read, write
      attribute:
      - givenname
      - displayname
      - title
      - initials
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory selfservice-present-copy.yml
```

其它资源

- 请参阅 IdM 中的自助服务访问控制。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-selfservice.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/selfservice 目录。

## 23.3. 使用 ANSIBLE 确保缺少自助服务规则

以下流程描述了如何使用 Ansible playbook 来确保 IdM 配置中没有指定的自助服务规则。以下示例描述了如何确保 Users can manage their own name details 自助服务规则在 IdM 中不存在。这将确保用户无法更改自己的显示名称或缩写。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/selfservice/ 目录中的 selfservice-absent.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-absent.yml selfservice-absent-copy.yml
```

3. 打开 selfservice-absent-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipaselfservice 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为自助服务规则的名称。
- 将 state 变量设置为 absent。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Self-service absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure self-service rule "Users can manage their own name details" is absent
    ipaselfservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "Users can manage their own name details"
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory selfservice-absent-copy.yml
```

**其它资源**

- 请参阅 IdM 中的自助服务访问控制。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-selfservice.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/selfservice 目录中的 playbook 示例。

## 23.4. 使用 ANSIBLE 确保自助服务规则具有特定属性

以下流程描述了如何使用 Ansible playbook 确保现有自助服务规则具有特定的设置。在示例中，您可以确认 Users can manage their own name details 自助服务规则也具有 surname 成员属性。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- Users can manage their own name details 自助服务规则存在于 IdM 中。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/selfservice/ 目录中的 selfservice-member-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-member-present.yml selfservice-member-present-copy.yml
```

3. 打开 selfservice-member-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipaselfservice 任务部分设置以下变量来调整文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
   - 将 name 变量设置为要修改的自助服务规则的名称。
   - 将 attribute 变量设置为 surname。
   - 将 action 变量设置为 member。

   这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Self-service member present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure selfservice "Users can manage their own name details" member
attribute surname is present
    ipaselfservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "Users can manage their own name details"
      attribute:
      - surname
      action: member
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory selfservice-
member-present-copy.yml
```

**其它资源**

- 请参阅 IdM 中的自助服务访问控制。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中提供的 README-selfservice.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/selfservice 目录中的 playbook 示例。

## 23.5. 使用 ANSIBLE 确保自助服务规则没有特定属性

以下流程描述了如何使用 Ansible playbook 来确保自助服务规则没有特定的设置。您可以使用此 playbook 确保自助服务规则没有授予不需要的访问权限。在示例中，您可以确定 Users can manage their own name details 自助服务规则没有包括 givenname 和 surname成员属性。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- Users can manage their own name details 自助服务规则存在于 IdM 中。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/selfservice/ 目录中的 selfservice-member-absent.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-member-absent.yml selfservice-member-absent-copy.yml
```

3. 打开 selfservice-member-absent-copy.yml Ansible playbook 文件进行编辑。

4. 通过在 ipaselfservice 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为您要修改的自助服务规则的名称。
- 将 attribute 变量设置为 givenname 和 top name。
- 将 action 变量设置为 member。
- 将 state 变量设置为 absent。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Self-service member absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure selfservice "Users can manage their own name details" member
attributes givenname and surname are absent
    ipaselfservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "Users can manage their own name details"
      attribute:
      - givenname
      - surname
      action: member
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory selfservice-
member-absent-copy.yml
```

**其它资源**

- 请参阅 IdM 中的自助服务访问控制。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-selfservice.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/selfservice 目录中的 playbook 示例。

# 第 24 章 将权限委派给用户组，来使用 IDM CLI 管理用户

委派是 IdM 中的访问控制方法之一，以及自助服务规则和基于角色的访问控制 (RBAC)。您可以使用委派（delegation）为一组用户分配权限，以管理另一组用户的条目。

本节涵盖了以下主题：

- 委派规则
- 使用 IdM CLI 创建委派规则
- 使用 IdM CLI 查看现有的委派规则
- 使用 IdM CLI 修改委派规则
- 使用 IdM CLI 删除委派规则

## 24.1. 委派规则

您可以通过创建委派规则，将权限委派给用户组来管理用户。

委派规则允许特定用户组对另一用户组中用户的特定属性执行写入（编辑）操作。这种形式的访问控制规则仅限于编辑您在委派规则中指定的属性子集的值；它不授予添加或删除整个条目或控制未指定属性的权限。

委派规则向 IdM 中的现有用户组授予权限。例如，您可以使用委派功能，允许 managers 用户组管理 employees 用户组中的选定用户属性。

## 24.2. 使用 IDM CLI 创建委派规则

按照以下流程，使用 IdM CLI 创建一个委派规则。

先决条件

- 您已作为 admins 组的成员登录。

流程

- 输入 ipa delegation-add 命令。指定以下选项：
  - --Group： *被授予用户组中用户条目权限* 的组。
  - --membergroup：*其条目可以被委派组的成员编辑* 的组。
  - --permissions ：用户是否有权查看给定属性（*读*），并添加或更改给定属性（*写*）。如果没有指定权限，则仅添加 *写* 权限。
  - --attrs ：允许成员组中的用户查看或编辑的属性。

  例如：

```
$ ipa delegation-add "basic manager attributes" --permissions=read --permissions=write --
attrs=businesscategory --attrs=departmentnumber --attrs=employeetype --
attrs=employeenumber --group=managers --membergroup=employees
----------------------------------------
Added delegation "basic manager attributes"
----------------------------------------
  Delegation name: basic manager attributes
  Permissions: read, write
  Attributes: businesscategory, departmentnumber, employeetype, employeenumber
  Member user group: employees
  User group: managers
```

### 24.3. 使用 IDM CLI 查看现有的委派规则

按照以下流程，使用 IdM CLI 查看现有的委派规则。

先决条件

- 您已作为 admins 组的成员登录。

流程

- 输入 ipa delegation-find 命令：

```
$ ipa delegation-find
-------------------
1 delegation matched
-------------------
  Delegation name: basic manager attributes
  Permissions: read, write
  Attributes: businesscategory, departmentnumber, employeenumber, employeetype
  Member user group: employees
  User group: managers
----------------------------
Number of entries returned 1
----------------------------
```

### 24.4. 使用 IDM CLI 修改委派规则

按照以下流程，使用 IdM CLI 修改现有的委派规则。

重要

--attrs 选项覆盖先前支持的属性列表，因此始终包括属性的完整列表以及任何新属性。这也适用于 --permissions 选项。

先决条件

- 您已作为 admins 组的成员登录。

流程

- 输入 ipa delegation-mod 命令及所需的更改。例如，要将 displayname 属性添加到 basic manager attributes 示例规则中：

```
$ ipa delegation-mod "basic manager attributes" --attrs=businesscategory --
attrs=departmentnumber --attrs=employeetype --attrs=employeenumber --
attrs=displayname
---------------------------------------------
Modified delegation "basic manager attributes"
```

```
----------------------------------------------
  Delegation name: basic manager attributes
  Permissions: read, write
  Attributes: businesscategory, departmentnumber, employeetype, employeenumber,
displayname
  Member user group: employees
  User group: managers
```

### 24.5. 使用 IDM CLI 删除委派规则

按照以下流程，使用 IdM CLI 删除现有的委派规则。

**先决条件**

- 您已作为 admins 组的成员登录。

**流程**

- 输入 ipa delegation-del 命令。
- 提示时，输入您要删除的委派规则的名称：

```
$ ipa delegation-del
Delegation name: basic manager attributes
----------------------------------------------
Deleted delegation "basic manager attributes"
----------------------------------------------
```

# 第 25 章 将权限委派给用户组，来使用 IDM WEB UI 管理用户

委派是 IdM 中的访问控制方法之一，以及自助服务规则和基于角色的访问控制 (RBAC)。您可以使用委派（delegation）为一组用户分配权限，以管理另一组用户的条目。

本节涵盖了以下主题：

- 委派规则
- 使用 IdM WebUI 创建委派规则
- 使用 IdM WebUI 查看现有的委派规则
- 使用 IdM WebUI 修改委派规则
- 使用 IdM WebUI 删除委派规则

## 25.1. 委派规则

您可以通过创建委派规则，将权限委派给用户组来管理用户。

委派规则允许特定用户组对另一用户组中用户的特定属性执行写入（编辑）操作。这种形式的访问控制规则仅限于编辑您在委派规则中指定的属性子集的值；它不授予添加或删除整个条目或控制未指定属性的权限。

委派规则向 IdM 中的现有用户组授予权限。例如，您可以使用委派功能，允许 managers 用户组管理 employees 用户组中的选定用户属性。

## 25.2. 使用 IDM WEBUI 创建委派规则

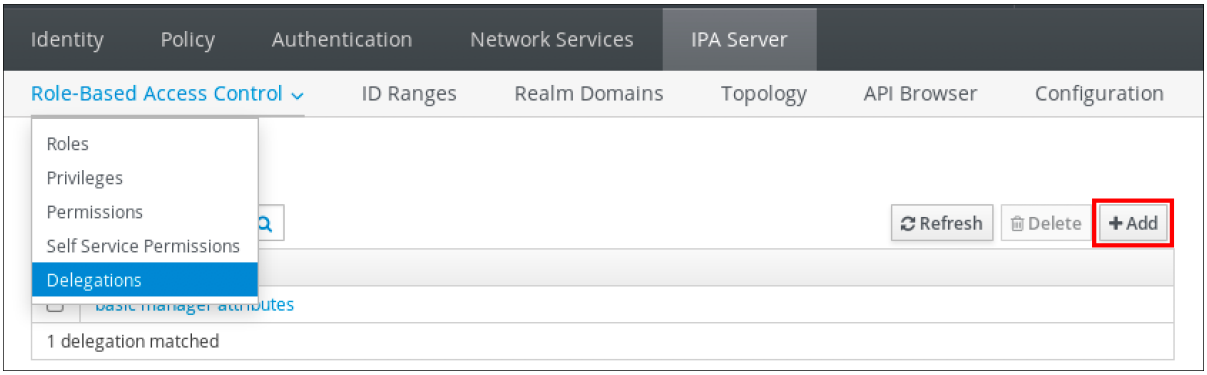
按照以下流程，使用 IdM WebUI 创建一个委派规则。

**先决条件**

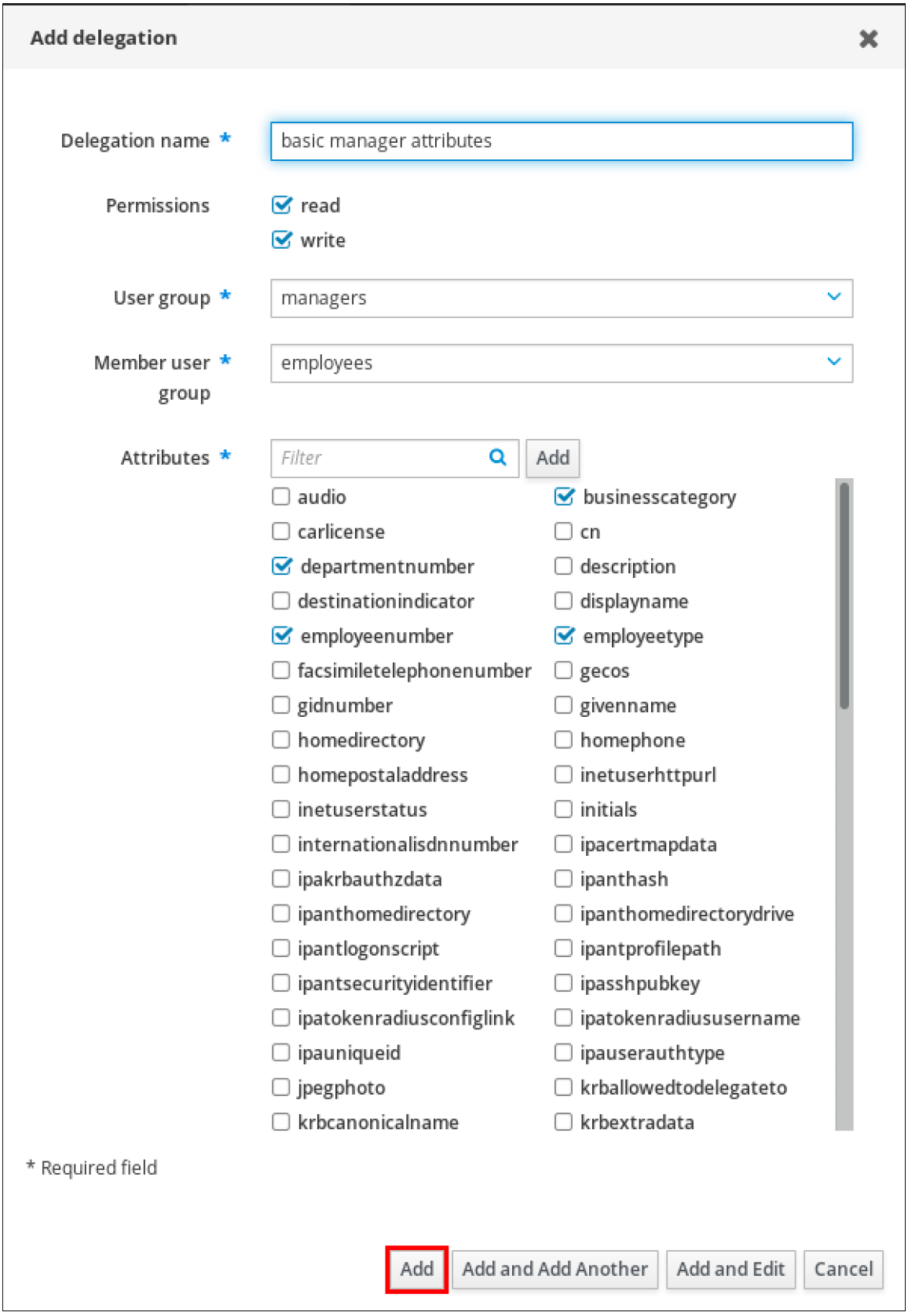
- 您已作为 admins 组的成员登录到 IdM Web UI。

**流程**

1. 在 IPA Server 菜单中点击 Role-Based Access Control → Delegations。
2. 点击 Add。




3. 在 Add delegation 窗口中执行以下操作：
   a. 命名新的委派规则。
   b. 通过选择复选框来设置权限，以指示用户是否有权查看给定的属性（*读*），并添加或更改给定的属性（*写*）。
   c. 在"用户组"下拉菜单中，选择 *被授予权限* 来查看或编辑成员组中的用户条目的组。
   d. 在 Member user group 下拉菜单中，选择*其条目可以被委派组的成员编辑* 的组。
   e. 在属性框中，按您要为其授予权限的属性选择复选框。




f. 单击 **Add** 按钮，以保存新的委派规则。

## 25.3. 使用 IDM WEBUI 查看现有的委派规则

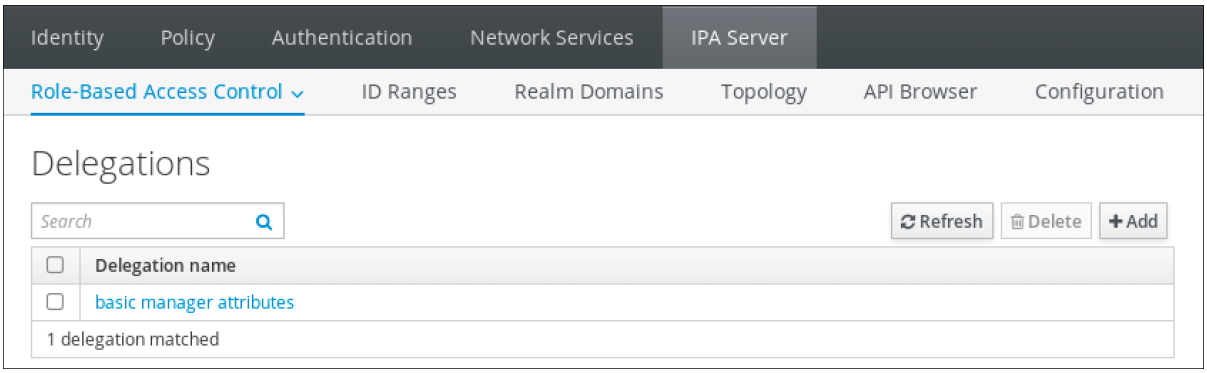
按照以下流程，使用 IdM WebUI 查看现有的委派规则。

**先决条件**

- 您已作为 admins 组的成员登录到 IdM Web UI。

**流程**

- 在 IPA Server 菜单中点击 Role-Based Access Control → Delegations。




## 25.4. 使用 IDM WEBUI 修改委派规则

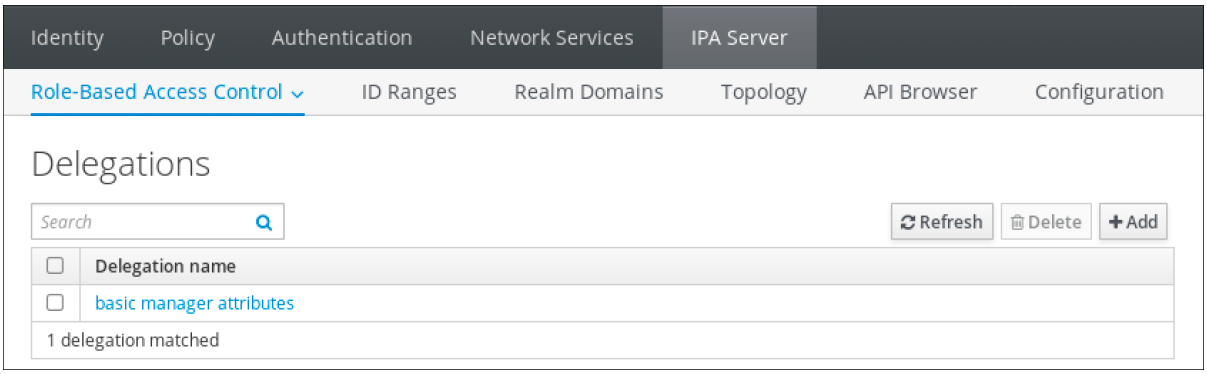
按照以下流程，使用 IdM Web UI 修改现有的委派规则。

**先决条件**

- 您已作为 admins 组的成员登录到 IdM Web UI。

**流程**

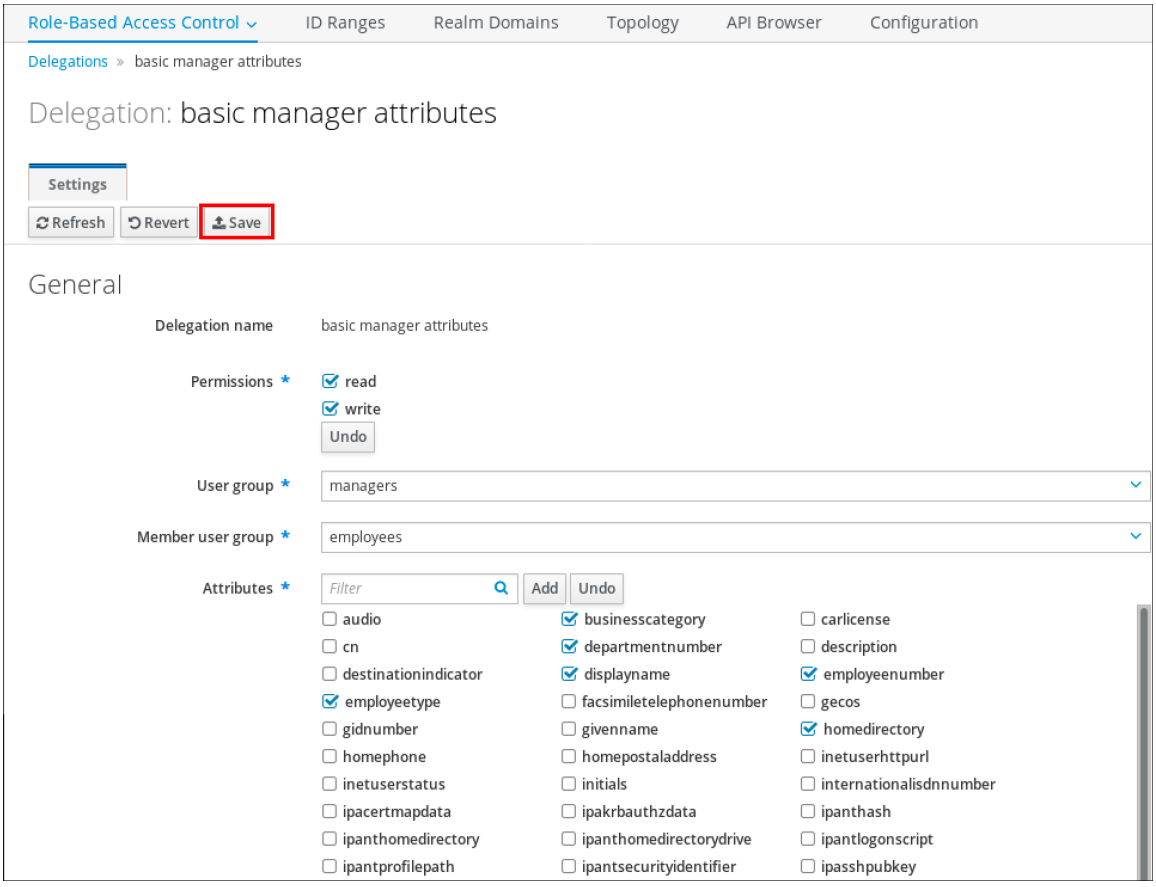
1. 在 IPA Server 菜单中点击 Role-Based Access Control → Delegations。



2. 点击您要修改的规则。

3. 进行所需的更改：

- 更改规则的名称。
- 通过选择复选框来更改授予的权限，这指示用户是否有权查看给定的属性（*读*），并添加或更改给定的属性（*写*）。
- 在"用户组"下拉菜单中，选择 *被授予权限* 来查看或编辑成员组中的用户条目的组。
- 在 **Member user group** 下拉菜单中，选择*其条目可以被委派组的成员编辑* 的组。
- 在属性框中，按您要为其授予权限的属性选择复选框。要删除对属性的权限，可取消相关的复选框。



- 单击 **Save** 按钮来保存更改。

## 25.5. 使用 IDM WEBUI 删除委派规则

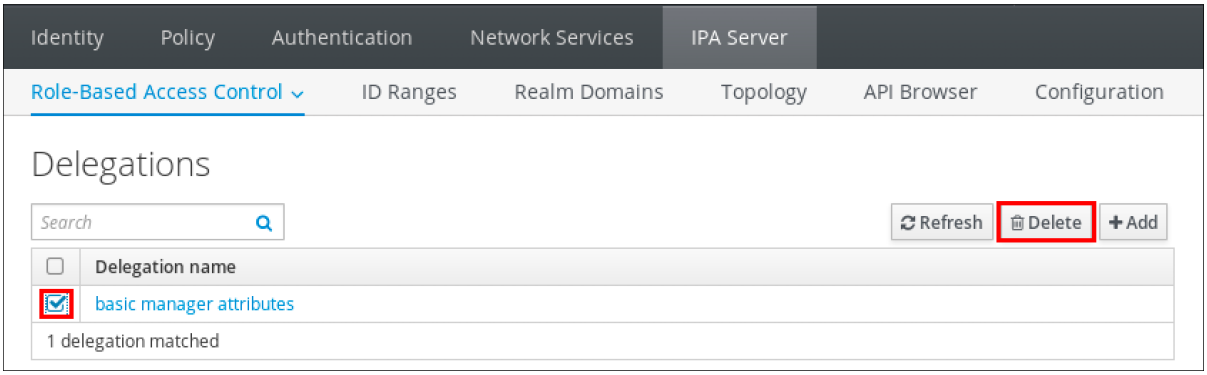
按照以下流程，使用 IdM Web UI 删除现有的委派规则。

### 先决条件

- 您已作为 **admins** 组的成员登录到 IdM Web UI。

### 流程

1. 在 **IPA Server** 菜单中点击 **Role-Based Access Control → Delegations**。
2. 选中您要删除的规则旁边的复选框。
3. 单击 **Delete**。




4. 单击 **Delete** 确认。

# 第 26 章 委派权限到用户组，以使用 ANSIBLE PLAYBOOK 管理用户

委派是 IdM 中的访问控制方法之一，以及自助服务规则和基于角色的访问控制 (RBAC)。您可以使用委派（delegation）为一组用户分配权限，以管理另一组用户的条目。

本节涵盖了以下主题：

- 委派规则
- 为 IdM 创建 Ansible 清单文件
- 使用 Ansible 确保存在委派规则
- 使用 Ansible 确保没有委派规则
- 使用 Ansible 确保委派规则具有特定属性
- 使用 Ansible 确保委派规则没有特定属性

## 26.1. 委派规则

您可以通过创建委派规则，将权限委派给用户组来管理用户。

委派规则允许特定用户组对另一用户组中用户的特定属性执行写入（编辑）操作。这种形式的访问控制规则仅限于编辑您在委派规则中指定的属性子集的值；它不授予添加或删除整个条目或控制未指定属性的权限。

委派规则向 IdM 中的现有用户组授予权限。例如，您可以使用委派功能，允许 managers 用户组管理 employees 用户组中的选定用户属性。

## 26.2. 为 IDM 创建 ANSIBLE 清单文件

在使用 Ansible 时，最好在主目录中创建一个专用于 Ansible playbook 的子目录，您可复制 /usr/share/doc/ansible-freeipa/* 和 /usr/share/doc/rhel-system-roles/* 子目录并进行相应的调整。这种做法有以下优点：

- 您可以在一个位置找到所有 playbook。
- 您可以运行 playbook，而无需调用 root 特权。

**流程**

1. 在主目录中为您的 Ansible 配置和 playbook 创建目录：

```
$ mkdir ~/MyPlaybooks/
```

2. 进入 ~/MyPlaybooks/ 目录：

```
$ cd ~/MyPlaybooks
```

3. 使用以下内容创建 ~/MyPlaybooks/ansible.cfg 文件：

```
[defaults]
inventory = /home/<username>/MyPlaybooks/inventory

[privilege_escalation]
become=True
```

4. 使用以下内容创建 ~/MyPlaybooks/inventory 文件：

```
[eu]
server.idm.example.com

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

此配置定义了两个主机组，即 eu 和 us，用于这些位置中的主机。此外，此配置定义了 ipaserver 主机组，它包含来自 eu 和 us 组的所有主机。

## 26.3. 使用 ANSIBLE 确保存在委派规则

以下流程描述了如何使用 Ansible playbook 为新的 IdM 委派规则定义特权并确保其存在。在这个示例中，新的 basic manager attributes 委派规则授予 managers 组为 employees 组成员读取和写入以下属性的权限：

- businesscategory
- departmentnumber
- employeenumber
- employeetype

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/delegation/ 目录中的 delegation-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-present.yml
delegation-present-copy.yml
```

3. 打开 delegation-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipadelegation 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为新委派规则的名称。
- 将 permission 变量设置为以逗号分隔的权限列表，以授予：read 和 write。
- 将 attribute 变量设置为委派的用户组可以管理的属性列表：businesscategory、departmentnumber、employeenumber 和 employeetype。
- 将 group 变量设置为被授予查看或修改属性访问权限的组名称。
- 将 membergroup 变量设置为组的名称，其属性可以查看或修改。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to manage a delegation rule
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure delegation "basic manager attributes" is present
    ipadelegation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "basic manager attributes"
      permission: read, write
      attribute:
      - businesscategory
      - departmentnumber
      - employeenumber
      - employeetype
      group: managers
      membergroup: employees
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
delegation-present-copy.yml
```

**其它资源**

- 请参阅 委派规则。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-delegation.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/ipadelegation 目录中的 playbook 示例。

## 26.4. 使用 ANSIBLE 确保没有委派规则

以下流程描述了如何使用 Ansible playbook 来确保您的 IdM 配置中没有指定的委托规则。以下示例描述了如何确保 IdM 中没有存在自定义 basic manager attributes 委派规则。

**先决条件**

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

流程

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks>/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/delegation/ 目录中的 delegation-absent.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-present.yml
delegation-absent-copy.yml
```

3. 打开 delegation-absent-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipadelegation 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为委派规则的名称。
- 将 state 变量设置为 absent。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Delegation absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure delegation "basic manager attributes" is absent
    ipadelegation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "basic manager attributes"
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
delegation-absent-copy.yml
```

**其它资源**

- 请参阅 委派规则。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-delegation.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/ipadelegation 目录中的 playbook 示例。

## 26.5. 使用 ANSIBLE 确保委派规则具有特定属性

以下流程描述了如何使用 Ansible playbook 确保委派规则具有特定的设置。您可以使用此 playbook 修改您之前创建的委派角色。在示例中，您可以确保 basic manager attributes 委派规则仅具有 departmentnumber 成员属性。

### 先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- IdM 中存在 basic manager attributes委派规则。

### 流程

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/delegation/ 目录中的 delegation-member-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-member-present.yml delegation-member-present-copy.yml
```

3. 打开 delegation-member-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipadelegation 任务部分设置以下变量来调整文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
   - 将 name 变量设置为要修改的委派规则的名称。
   - 将 attribute 变量设置为 departmentnumber。
   - 将 action 变量设置为 member。

   这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Delegation member present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure delegation "basic manager attributes" member attribute departmentnumber is present
    ipadelegation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "basic manager attributes"
      attribute:
      - departmentnumber
      action: member
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

   -

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
delegation-member-present-copy.yml
```

其它资源

- 请参阅 委派规则。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-delegation.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/ipadelegation 目录中的 playbook 示例。

## 26.6. 使用 ANSIBLE 确保委派规则没有特定属性

以下流程描述了如何使用 Ansible playbook 来确保委派规则没有特定的设置。您可以使用此 playbook 确保委派角色不授予不需要的访问权限。在该示例中，您可以确保 basic manager attributes 委派规则没有 employeenumber 和 employeetype 成员属性。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、

服务器或副本的一部分。

- IdM 中存在 basic manager attributes委派规则。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/delegation/ 目录中的 delegation-member-absent.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-member-absent.yml delegation-member-absent-copy.yml
```

3. 打开 delegation-member-absent-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipadelegation 任务部分设置以下变量来调整文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
   - 将 name 变量设置为要修改的委派规则的名称。
   - 将 attribute 变量设置为 employeenumber 和 employeetype。
   - 将 action 变量设置为 member。
   - 将 state 变量设置为 absent。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Delegation member absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure delegation "basic manager attributes" member attributes
employeenumber and employeetype are absent
    ipadelegation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "basic manager attributes"
      attribute:
      - employeenumber
      - employeetype
      action: member
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
delegation-member-absent-copy.yml
```

**其它资源**

- 请参阅 委派规则。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-delegation.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/ipadelegation 目录中的 playbook 示例。

# 第 27 章 使用 CLI 在 IDM 中管理基于角色的访问控制

了解有关身份管理(IdM)中的基于角色的访问控制以及命令行界面(CLI)中运行的以下操作的更多信息：

- 管理权限
- 管理特权
- 管理角色

## 27.1. IDM 中的基于角色的访问控制

与自助服务和委派访问控制相比，IdM 中的基于角色的访问控制(RBAC)向用户授予了完全不同的权限。

基于角色的访问控制由三个部分组成：

- Permissions 授予执行特定任务的权利，如添加或删除用户、修改组和启用读权限。
- Privileges（特权） 结合了权限，例如添加新用户所需的所有权限。
- Roles（角色） 向用户、用户组、主机或主机组授予一组特权。

### 27.1.1. IdM 中的权限

权限是基于角色的访问控制的最低级别单元，它们定义这些操作所应用到的 LDAP 条目。与构建块类似，可以根据需要将权限分配给多个特权。
一个或多个权利定义了允许的操作 ：

- write

- 读取
- search
- compare
- add
- delete
- all

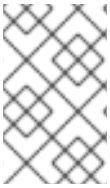
这些操作适用于三个基本目标：

- subtree：域名 (DN)；此 DN 下的子树
- target filter：LDAP 过滤器
- target：可以带有通配符的 DN 指定条目

此外，以下方便选项可设置对应的属性：

- type：对象类型（用户、组等）；设置subtree 和 target filter
- memberof：组成员；设置 target filter
- targetgroup：授予修改特定组的权限（如授予管理组成员资格的权限）；设置 target

使用 IdM 权限，您可以控制哪些用户有权访问哪些对象，甚至控制这些对象的属性。IdM 允许您允许或阻止单个属性，或更改特定 IdM 功能（如用户、组或 sudo）的所有可见性，适用于所有匿名用户、所有经过身份验证的用户，或者只更改一组特定的特权用户。
例如，如果管理员只想将用户或组的访问权限限制到这些用户或组需要访问的特定部分，并且使其他部分完全隐藏于他们，此方法的灵活性对管理员很有用。



注意

权限不能包含其他权限。

### 27.1.2. 默认管理的权限

管理的权限是 IdM 默认附带的权限。它们的行为与用户创建的其他权限类似，但有以下区别：

- 您无法删除它们或修改其名称、位置和目标属性。
- 它们有三组属性：
  - Default 属性，用户无法修改它们，因为它们由 IdM 管理
  - Included 属性，它们是用户添加的额外属性
  - Excluded 属性，这些属性由用户删除

管理的权限适用于 default 和 included 属性集中显示的所有属性，但不应用到排除集中的所有属性。

注意

虽然您无法删除受管权限，但将其绑定类型设置为权限并从所有特权中删除托管权限会有效地禁用该权限。

所有受管权限的名称都以 System: 开头，例如 System: Add Sudo rule 或 System: Modify Services。IdM 的早期版本将不同的方案用于默认权限。例如，用户无法删除它们，而只能将它们分配到特权。这些默认权限大部分已转换为受管权限，但以下权限仍使用以前的方案：

- 添加自动成员重新构建成员身份任务
- 添加配置子条目
- 添加复制协议
- 证书删除冻结
- 从 CA 获取证书状态
- 读取 DNA 范围
- 修改 DNA 范围
- 读取 PassSync Manager 配置
- 修改 PassSync Manager 配置
- 阅读复制协议
- 修改复制协议
- 删除复制协议
- 读取 LDBM 数据库配置
- 请求证书
-

- 请求证书忽略 CA ACL
- 从不同主机请求证书
- 从 CA 检索证书
- 吊销证书
- 写入 IPA 配置

**注意**

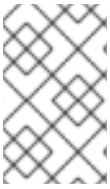
如果您试图通过命令行修改受管权限，系统不允许更改您无法修改的属性，命令会失败。如果您试图从 Web UI 修改受管权限，则无法修改的属性将被禁用。

### 27.1.3. IdM 中的特权

特权是一组适用于角色的权限。
虽然权限提供了执行单个操作的权限，但某些 IdM 任务需要多个权限才能成功。因此，特权组合了执行特定任务所需的不同权限。
例如，为新 IdM 用户设置帐户需要以下权限：

- 创建新用户条目
- 重置用户密码
- 将新用户添加到默认 IPA 用户组

将这三个低级别任务合并到一个更高级别的任务中，例如名为 Add User，可使系统管理员更加轻松地管理角色。IdM 已包含几个默认权限。除了用户和用户组外，还将特权分配到主机和主机组，以及网络服务。这种方法允许精细控制一组主机上使用特定网络服务的操作。

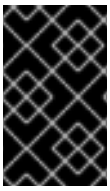


**注意**

特权可能不包含其他特权。

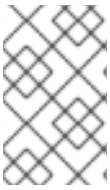
### 27.1.4. IdM 中的角色

角色是用户为角色指定的特权列表。
实际上，权限授予执行给定低级别任务（如创建用户条目和向组添加一个条目）的能力，特权将高级别任务所需的一个或多个这些权限（如在给定组中创建新用户）组合在一起。角色根据需要收集权限：例如，用户管理员角色能够添加、修改和删除用户。



**重要**

角色用于对允许的操作进行分类。它们不用作实施特权升级或防止特权升级的工具。



**注意**

角色不能包含其他角色。

### 27.1.5. Identity Management 中的预定义角色

Red Hat Identity Management 提供以下预定义角色范围：

**表 27.1. 身份管理中的预定义角色**

| 角色 | 特权 | 描述 |
|---|---|---|
| Enrollment Administrator | 主机注册 | 负责客户端或主机、注册 |
| helpdesk | 改用户和重置密码，修改组成员身份 | 负责执行简单的用户管理任务 |
| IT Security Specialist | Netgroups 管理员, HBAC 管理员, Sudo 管理员 | 负责管理安全策略，如基于主机的访问控制、sudo 规则 |
| IT Specialist | 主机管理员、主机组管理员、服务管理员、自动装载管理员 | 负责管理主机 |
| Security Architect | 委派管理员、复制管理员、写 IPA 配置、密码策略管理员 | 负责管理身份管理环境、创建信任、创建复制协议 |

| 角色 | 特权 | 描述 |
| --- | --- | --- |
| User Administrator | 用户管理员、组管理员、阶段用户管理员 | 负责创建用户和组 |

## 27.2. 在 CLI 中管理 IDM 权限

按照以下流程，使用命令行界面(CLI)管理身份管理(IdM)权限。

### 先决条件

- 管理 IdM 或 用户管理员 角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。

### 流程

1. 使用 ipa permission-add 命令创建新的权限条目。
例如，添加名为 *dns admin* 的权限：

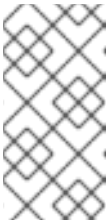
```
$ ipa permission-add "dns admin"
```

2. 使用以下选项指定权限的属性：

- --bindtype 指定绑定规则类型。此选项接受 all、anonymous 和 permission 参数。permission bindtype 表示只有通过角色授予了此权限的用户才能执行它。
例如：

```
$ ipa permission-add "dns admin" --bindtype=all
```

如果没有指定 --bindtype，则 permission 是默认值。



注意

不能对特权添加带有非默认绑定规则类型的权限。您也不能对非默认绑定规则类型设置特权中已存在的权限。

- --right 列出了权限授予的权力，它替换了已弃用的 --permissions 选项。可用的值有 add、delete、read、search、compare、write、all。

  您可以使用多个 --right 选项或使用大括号内以逗号分隔的列表来设置多个属性。例如：

```
$ ipa permission-add "dns admin" --right=read --right=write
```

```
$ ipa permission-add "dns admin" --right={read,write}
```

注意

add 和 delete 是入门级操作（例如，删除用户、添加组等），而 read、search、compare 和 write 是属性级别操作：您可以写入 userCertificate 而不是读 userPassword。

- --attrs 提供被授予权限的属性列表。
  您可以使用多个 --attrs 选项或通过在大括号内以逗号分隔的列表列出选项，来设置多个属性。例如：

```
$ ipa permission-add "dns admin" --attrs=description --attrs=automountKey
```

```
$ ipa permission-add "dns admin" --attrs={description,automountKey}
```

  使用 --attrs 提供的属性必须存在，并且是给定对象类型的允许属性，否则命令会失败，并显示模式语法错误。

- --type 定义对其应用权限的条目对象类型，如用户、主机或服务。每种类型都有其自己的一组允许的属性。
  例如：

```
$ ipa permission-add "manage service" --right=all --type=service --
attrs=krbprincipalkey --attrs=krbprincipalname --attrs=managedby
```

- --subtree 提供子树条目；然后，过滤器以这个子树条目下的每个条目为目标。提供现有

的子树条目；--subtree 不接受通配符或不存在的域名(DN)。在目录中包括 DN。
因为 IdM 使用简化的扁平目录树结构，所以 --subtree 可用于将某些类型的条目作为目标，如自动挂载位置，它们是其他配置的容器或父条目。例如：

```
$ ipa permission-add "manage automount locations" --
subtree="ldap://ldap.example.com:389/cn=automount,dc=example,dc=com" --
right=write --attrs=automountmapname --attrs=automountkey --
attrs=automountInformation
```

注意

--type 和 --subtree 选项是互斥的：您可以将 --type 包含的过滤器视为 --subtree 的简化，目的是使管理员的工作更为简单。

- --filter 使用 LDAP 过滤器来识别权限应用到哪个条目。
IdM 自动检查给定过滤器的有效性。过滤器可以是任何有效的 LDAP 过滤器，例如：

```
$ ipa permission-add "manage Windows groups" --filter="(!
(objectclass=posixgroup))" --right=write --attrs=description
```

- 检查组是否存在后，--memberof 对给定组的成员设置目标过滤器。例如，要让拥有此权限的用户修改 engineers 组成员的登录 shell：

```
$ ipa permission-add ManageShell --right="write" --type=user --attr=loginshell --
memberof=engineers
```

- 在检查组存在后，--targetgroup 对指定的用户组设置目标。例如，要让那些在 engineers 组中的人拥有写成员属性的权限（这样他们可以添加或删除成员）：

```
$ ipa permission-add ManageMembers --right="write" --
subtree=cn=groups,cn=accounts,dc=example,dc=test --attr=member --
targetgroup=engineers
```

- 另外，您还可以指定目标域名(DN)：
  - --target 指定要对其应用权限的 DN。可接受通配符.
  - --targetto 指定条目可移动到的 DN 子树。

- --targetfrom 指定可从中移出条目的 DN 子树。

## 27.3. 现有权限的命令选项

根据需要，使用以下变体修改现有权限：

- 要编辑现有权限，请使用 ipa permission-mod 命令。您可以使用与添加权限相同的命令选项。
- 要查找现有权限，请使用 ipa permission-find 命令。您可以使用与添加权限相同的命令选项。
- 要查看特定的权限，请使用 ipa permissions-show 命令。--raw 参数显示生成的原始 389-ds ACI。例如：

```
$ ipa permission-show <permission> --raw
```

- ipa permissions-del 命令完全删除权限。

其它资源

- 请参阅 ipa man page。
- 请参阅 ipa help 命令。

## 27.4. 在 CLI 中管理 IDM 特权

按照以下流程，使用命令行界面(CLI)管理身份管理(IdM)特权。

先决条件

- 管理 IdM 或 用户管理员 角色的管理员特权。

- 一个活跃的 Kerberos 票据。详情请查看 link: 使用 kinit 手动登录到 IdM。
- 现有权限。有关权限的详情，请参阅 在 CLI 中管理 IdM 权限。

**流程**

1. 使用 ipa privilege-add 命令添加权限条目，
例如，添加名为 *managing filesystems* 的特权并带有描述：

```
$ ipa privilege-add "managing filesystems" --desc="for filesystems"
```

2. 使用 privilege-add-permission 命令为特权组分配所需的权限，
例如，将名为 *managing automount* 和 *managing ftp services* 的权限添加到 *managing filesystems* 特权 ：

```
$ ipa privilege-add-permission "managing filesystems" --permissions="managing automount" --permissions="managing ftp services"
```

## 27.5. 现有权限的命令选项

根据需要，使用以下变体修改现有特权：

- 若要修改现有特权，可使用 ipa privilege-mod 命令。
- 要查找现有特权，请使用 ipa privilege-find 命令。
- 若要查看特定的特权，可使用 ipa privilege-show 命令。
- ipa privilege-remove-permission 命令从特权中删除一个或多个权限。
- ipa privilege-del 命令完全删除特权。

**其它资源**

- 请参阅 ipa man page。
- 请参阅 ipa help 命令。

### 27.6. 在 CLI 中管理 IDM 角色

按照以下流程，使用命令行界面(CLI)管理身份管理(IdM)角色。

**先决条件**

- 管理 IdM 或 用户管理员 角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。
- 现有特权。有关特权的详情，请参阅 在 CLI 中管理 IdM 特权。

**流程**

1. 使用 ipa role-add 命令添加新角色条目：

```
$ ipa role-add --desc="User Administrator" useradmin
-----------------------
Added role "useradmin"
-----------------------
Role name: useradmin
Description: User Administrator
```

2. 使用 ipa role-add-privilege 命令将所需的特权添加到角色中：

```
$ ipa role-add-privilege --privileges="user administrators" useradmin
Role name: useradmin
Description: User Administrator
Privileges: user administrators
---------------------------
Number of privileges added 1
---------------------------
```

3. 使用 ipa role-add-member 命令将所需的成员添加到角色中。允许的成员类型有：users、

groups、hosts hostgroups。
例如，将名为 *useradmins* 的组添加到之前创建的 *useradmin* 角色中：

```
$ ipa role-add-member --groups=useradmins useradmin
Role name: useradmin
Description: User Administrator
Member groups: useradmins
Privileges: user administrators
-------------------------
Number of members added 1
-------------------------
```

## 27.7. 现有角色的命令选项

根据需要，使用以下变体修改现有角色：

- 若要修改现有角色，请使用 ipa role-mod 命令。
- 要查找现有角色，请使用 ipa role-find 命令。
- 要查看特定的角色，请使用 ipa role-show 命令。
- 若要从角色中删除成员，请使用 ipa role-remove-member 命令。
- ipa role-remove-privilege 命令从角色中删除一个或多个特权。
- ipa role-del 命令将完全删除角色。

其它资源

- 请参阅 ipa 手册页
- 请参阅 ipa help 命令。

# 第 28 章 使用 IDM WEB UI 管理基于角色的访问控制

了解有关身份管理(IdM)中的基于角色的访问控制以及在 Web 界面(Web UI)中运行的以下操作的更多信息：

- 管理权限
- 管理特权
- 管理角色

## 28.1. IDM 中的基于角色的访问控制

与自助服务和委派访问控制相比，IdM 中的基于角色的访问控制(RBAC)向用户授予了完全不同的权限。

基于角色的访问控制由三个部分组成：

- Permissions 授予执行特定任务的权利，如添加或删除用户、修改组和启用读权限。
- Privileges（特权） 结合了权限，例如添加新用户所需的所有权限。
- Roles（角色） 向用户、用户组、主机或主机组授予一组特权。

### 28.1.1. IdM 中的权限

权限是基于角色的访问控制的最低级别单元，它们定义这些操作所应用到的 LDAP 条目。与构建块类似，可以根据需要将权限分配给多个特权。
一个或多个权利定义了允许的操作 ：

- write

- 读取
- search
- compare
- add
- delete
- all

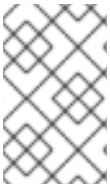
这些操作适用于三个基本目标：

- subtree：域名 (DN)；此 DN 下的子树
- target filter：LDAP 过滤器
- target：可以带有通配符的 DN 指定条目

此外，以下方便选项可设置对应的属性：

- type：对象类型（用户、组等）；设置subtree 和 target filter
- memberof：组成员；设置 target filter
- targetgroup：授予修改特定组的权限（如授予管理组成员资格的权限）；设置 target

使用 IdM 权限，您可以控制哪些用户有权访问哪些对象，甚至控制这些对象的属性。IdM 允许您允许或阻止单个属性，或更改特定 IdM 功能（如用户、组或 sudo）的所有可见性，适用于所有匿名用户、所有经过身份验证的用户，或者只更改一组特定的特权用户。
例如，如果管理员只想将用户或组的访问权限限制到这些用户或组需要访问的特定部分，并且使其他部分完全隐藏于他们，此方法的灵活性对管理员很有用。



注意

权限不能包含其他权限。

### 28.1.2. 默认管理的权限

管理的权限是 IdM 默认附带的权限。它们的行为与用户创建的其他权限类似，但有以下区别：

- 您无法删除它们或修改其名称、位置和目标属性。
- 它们有三组属性：
  - Default 属性，用户无法修改它们，因为它们由 IdM 管理
  - Included 属性，它们是用户添加的额外属性
  - Excluded 属性，这些属性由用户删除

管理的权限适用于 default 和 included 属性集中显示的所有属性，但不应用到排除集中的所有属性。

注意

虽然您无法删除受管权限，但将其绑定类型设置为权限并从所有特权中删除托管权限会有效地禁用该权限。

所有受管权限的名称都以 System: 开头，例如 System: Add Sudo rule 或 System: Modify Services。IdM 的早期版本将不同的方案用于默认权限。例如，用户无法删除它们，而只能将它们分配到特权。这些默认权限大部分已转换为受管权限，但以下权限仍使用以前的方案：

- 添加自动成员重新构建成员身份任务
- 添加配置子条目
- 添加复制协议
- 证书删除冻结
- 从 CA 获取证书状态
- 读取 DNA 范围
- 修改 DNA 范围
- 读取 PassSync Manager 配置
- 修改 PassSync Manager 配置
- 阅读复制协议
- 修改复制协议
- 删除复制协议
- 读取 LDBM 数据库配置
- 请求证书
-

- 请求证书忽略 CA ACL
- 从不同主机请求证书
- 从 CA 检索证书
- 吊销证书
- 写入 IPA 配置

**注意**

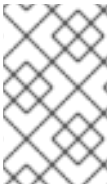
如果您试图通过命令行修改受管权限，系统不允许更改您无法修改的属性，命令会失败。如果您试图从 Web UI 修改受管权限，则无法修改的属性将被禁用。

### 28.1.3. IdM 中的特权

特权是一组适用于角色的权限。
虽然权限提供了执行单个操作的权限，但某些 IdM 任务需要多个权限才能成功。因此，特权组合了执行特定任务所需的不同权限。
例如，为新 IdM 用户设置帐户需要以下权限：

- 创建新用户条目
- 重置用户密码
- 将新用户添加到默认 IPA 用户组

将这三个低级别任务合并到一个更高级别的任务中，例如名为 Add User，可使系统管理员更加轻松地管理角色。IdM 已包含几个默认权限。除了用户和用户组外，还将特权分配到主机和主机组，以及网络服务。这种方法允许精细控制一组主机上使用特定网络服务的操作。

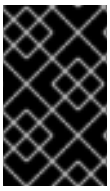


**注意**

特权可能不包含其他特权。

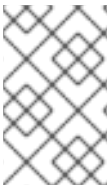
### 28.1.4. IdM 中的角色

角色是用户为角色指定的特权列表。
实际上，权限授予执行给定低级别任务（如创建用户条目和向组添加一个条目）的能力，特权将高级别任务所需的一个或多个这些权限（如在给定组中创建新用户）组合在一起。角色根据需要收集权限：例如，用户管理员角色能够添加、修改和删除用户。



**重要**

角色用于对允许的操作进行分类。它们不用作实施特权升级或防止特权升级的工具。



**注意**

角色不能包含其他角色。

### 28.1.5. Identity Management 中的预定义角色

Red Hat Identity Management 提供以下预定义角色范围：

**表 28.1. 身份管理中的预定义角色**

| 角色 | 特权 | 描述 |
|---|---|---|
| Enrollment Administrator | 主机注册 | 负责客户端或主机、注册 |
| helpdesk | 改用户和重置密码，修改组成员身份 | 负责执行简单的用户管理任务 |
| IT Security Specialist | Netgroups 管理员, HBAC 管理员, Sudo 管理员 | 负责管理安全策略，如基于主机的访问控制、sudo 规则 |
| IT Specialist | 主机管理员、主机组管理员、服务管理员、自动装载管理员 | 负责管理主机 |
| Security Architect | 委派管理员、复制管理员、写 IPA 配置、密码策略管理员 | 负责管理身份管理环境、创建信任、创建复制协议 |

| 角色 | 特权 | 描述 |
| --- | --- | --- |
| User Administrator | 用户管理员、组管理员、阶段用户管理员 | 负责创建用户和组 |

### 28.2. 在 IDM WEB UI 中管理权限

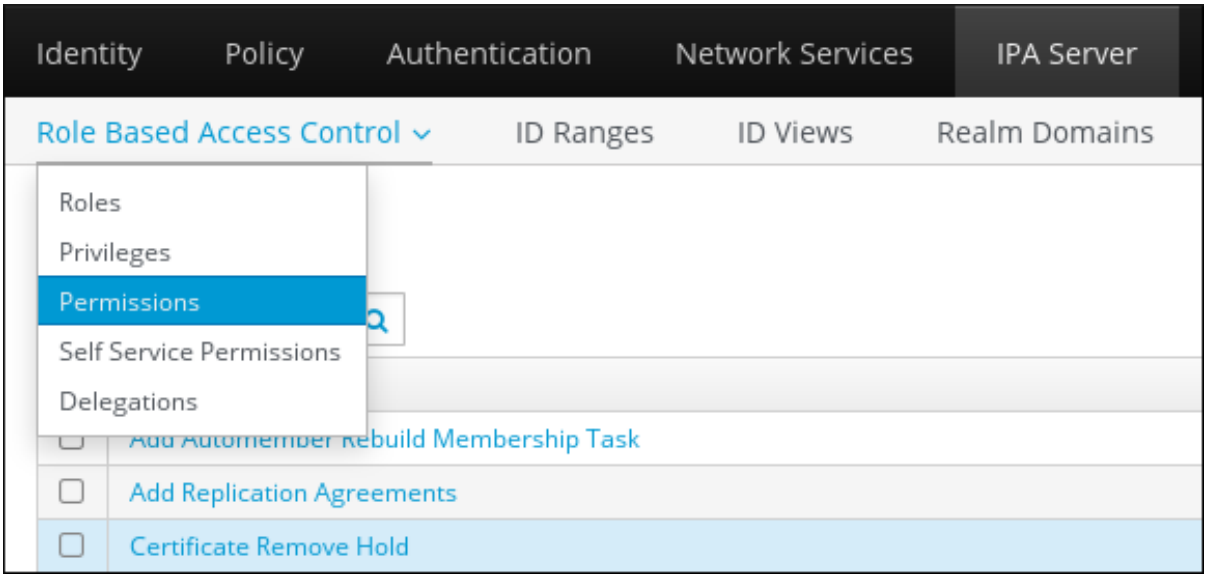
按照以下流程，使用 Web 界面(IdM Web UI)在身份管理(IdM)中管理权限。

**先决条件**

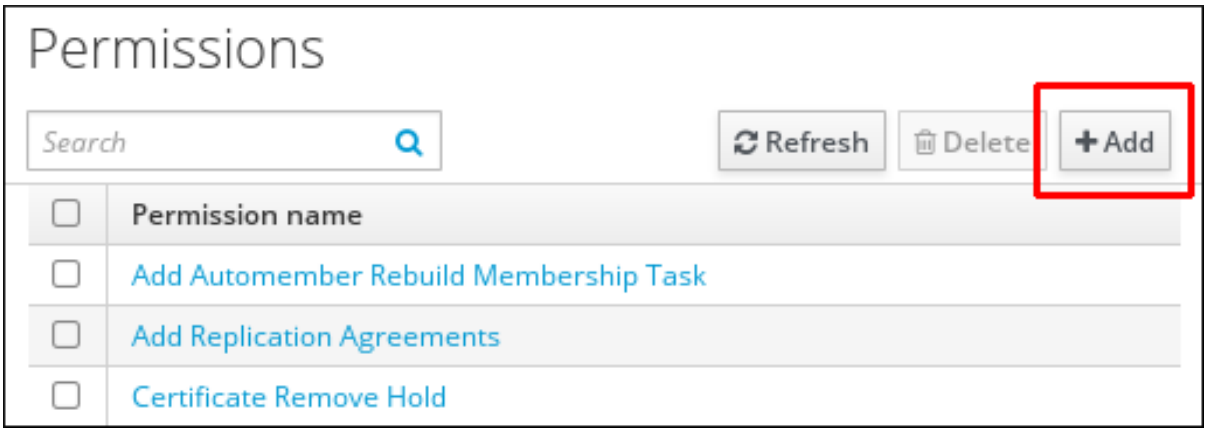
- 管理 IdM 或 用户管理员 角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

**流程**

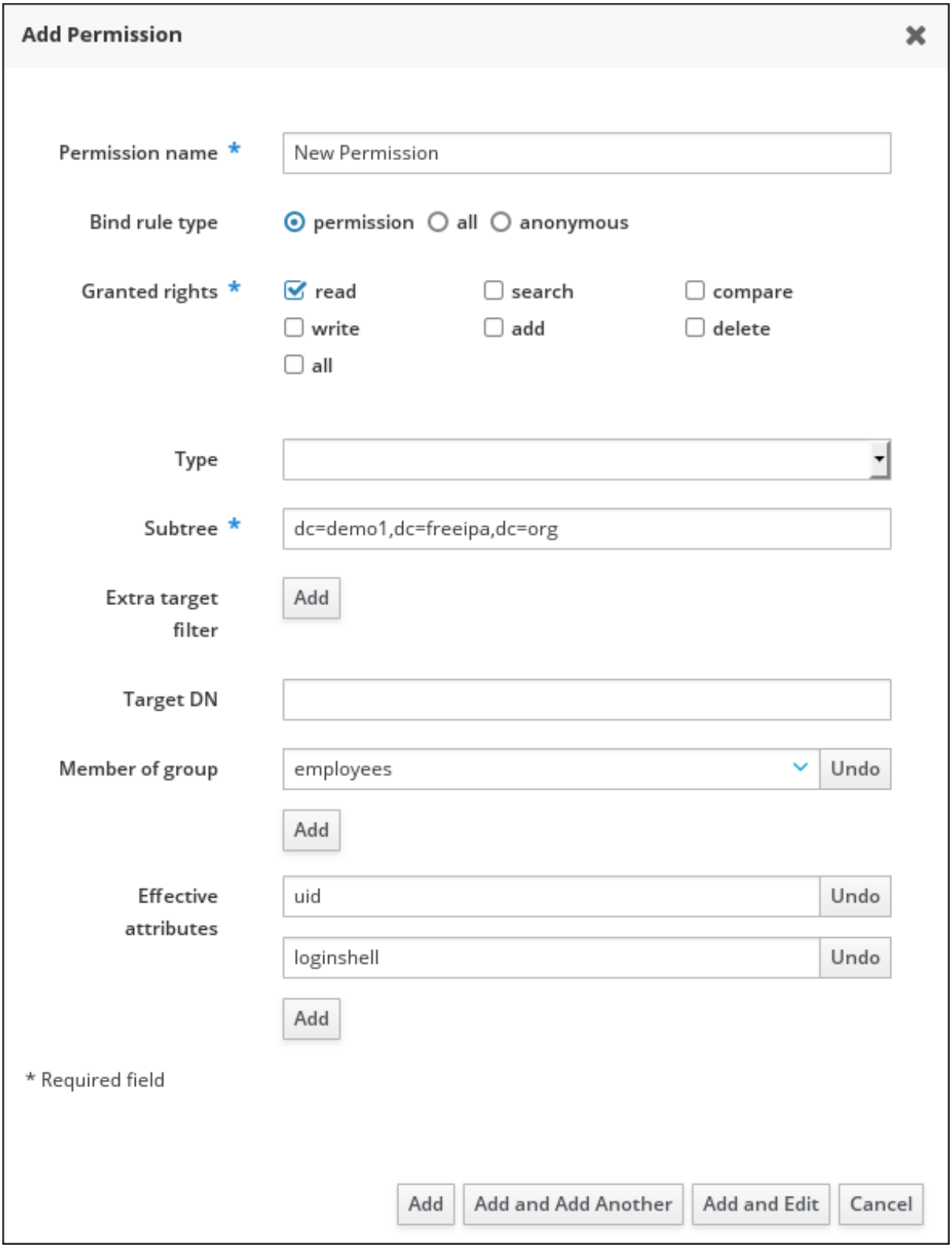
1. 要添加一个新权限，请在 IPA Server 选项卡中打开 Role-Based Access Control 子菜单，然后选择 Permissions ：




2. 此时会打开权限列表：点击权限列表顶部的 Add 按钮：




3. **此时会打开 Add Permission 表单。指定新权限的名称，并相应地定义其属性：**




4. 选择合适的绑定规则类型：

- permission 是默认的权限类型，通过特权和角色授予访问权限

- all 指定权限适用于所有经过身份验证的用户

- anonymous 指定权限适用于所有用户，包括未经身份验证的用户

**注意**

不能对特权添加带有非默认绑定规则类型的权限。您也不能对非默认绑定规则类型设置特权中已存在的权限。

5. 选择在 Granted rights 中使用此权限授予的权利。

6. 定义方法来标别权限的目标条目：

- Type 指定条目类型，如 user、host 或 service。如果您为 Type 设置选择了一个值，则可通过该 ACI 访问该条目类型的所有可能属性的列表将出现在 Effective Attributes 下。定义 Type 会将 Subtree 和 Target DN 设置为其中一个预定义的值。

- Subtree （必需的）指定一个子树条目；然后这个子树条目下的每个条目都成为目标。提供现有的子树条目，因为 Subtree 不接受通配符或不存在的域名(DN)。例如：cn=automount,dc=example,dc=com

- 额外目标过滤器 使用 LDAP 过滤器来识别权限将应用到哪个条目。过滤器可以是任何有效的 LDAP 过滤器，例如： (!(objectclass=posixgroup))
，IdM 会自动检查给定过滤器的有效性。如果您输入无效的过滤器，IdM 会在您尝试保存权限时给您发出警告。

- 目标 DN 指定域名(DN)，并接受通配符。例如：uid=*,cn=users,cn=accounts,dc=com

- 组成员 对给定组的成员设置目标过滤器。指定过滤器设置并点击 Add 后，IdM 会验证过滤器。如果所有权限设置都正确，IdM 将执行搜索。如果某些权限设置不正确，IdM 将显示一条消息，通知您哪个设置不正确。

7. 向权限添加属性：

- 如果设置了 Type，请从可用的 ACI 属性列表中选择 Effective attributes。

- 如果您没有使用 Type，通过将属性写入Effective attributes 字段来手动添加属性。一次添加一个属性；若要添加多个属性，可单击 Add 来添加另一个输入字段。

重要

如果您没有为权限设置任何属性，则权限默认包含所有属性。

8. 使用表单底部的 Add 按钮完成添加权限：

- 单击 Add 按钮来保存权限，并回到权限列表。

- 或者，您可以保存权限，并通过单击 Add and Add another 按钮继续在同一表单中添加其他权限。

- Add and Edit 按钮使您可以保存并继续编辑新创建的权限。

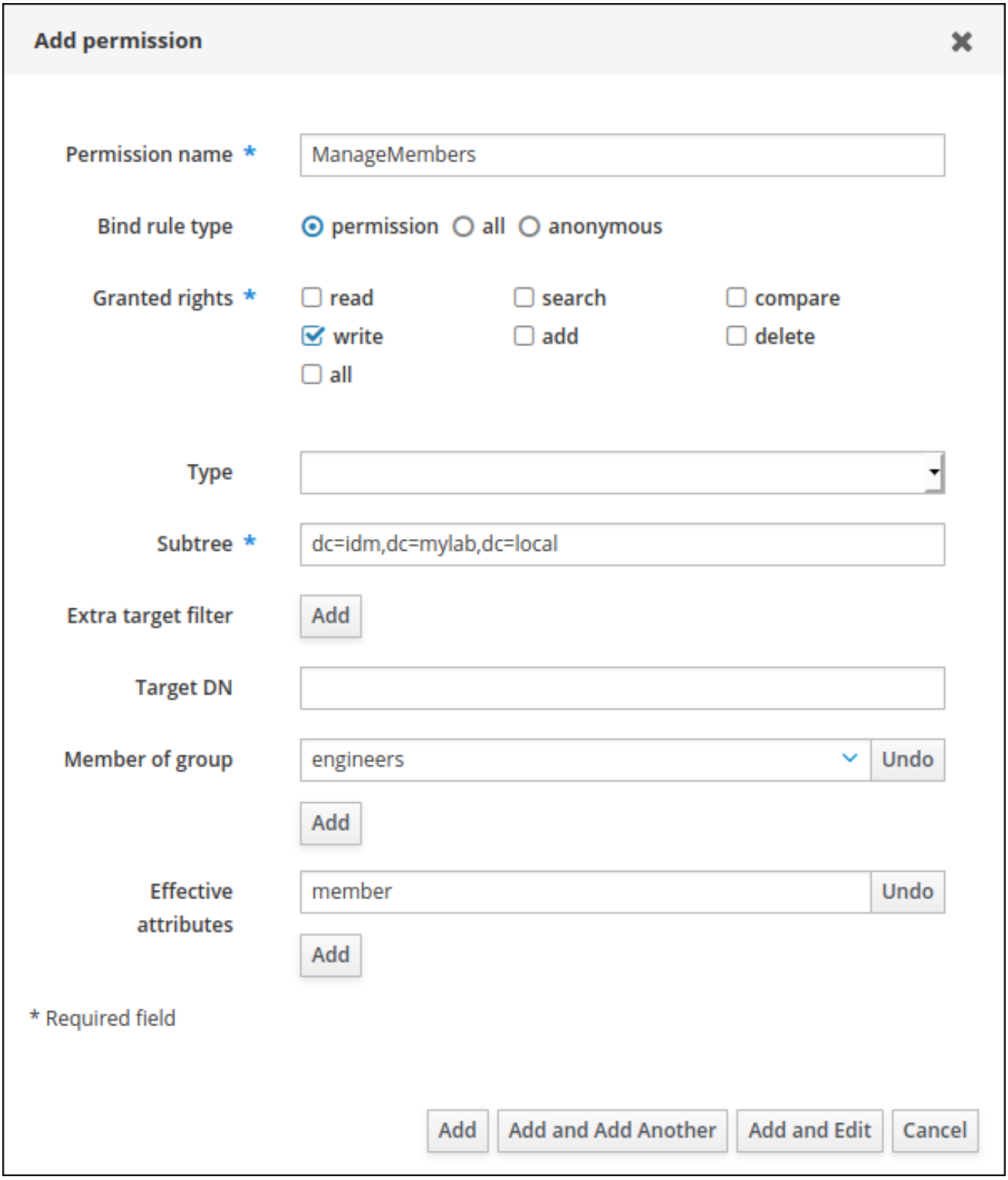
9. *可选。*您还可以通过单击权限列表中的名称来显示Permission settings 页面来编辑现有权限的属性。

10. *可选。*如果您需要删除现有权限，请在列表中选中其名称旁边的复选框后单击 Delete按钮，来显示 Remove permissions 对话框。

注意

对默认受管权限的操作是受限制的：您无法修改的属性在 IdM Web UI 中是禁用的，您无法完全删除受管的权限。
但是，您可以通过从所有特权中删除受管权限，可以有效禁用设置了绑定类型权限的受管权限。

例如，要让 engineer 组中的用户拥有写成员属性的权限（因此他们可以添加或删除成员）：




### 28.3. 在 IDM WEB UI 中管理特权

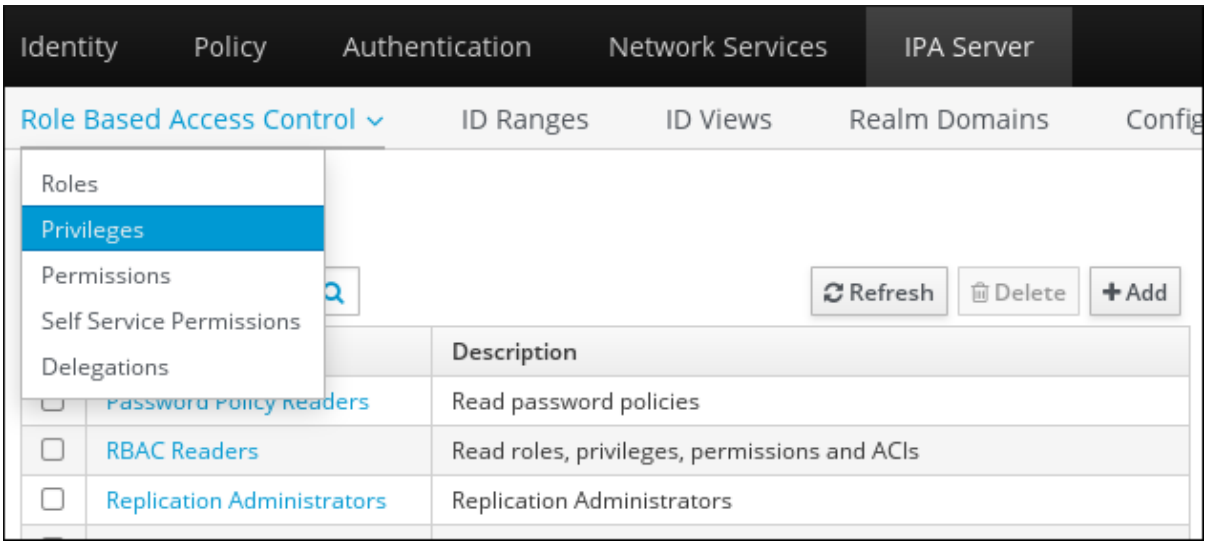
按照以下流程，使用 Web 界面(IdM Web UI)在 IdM 中管理特权。

**先决条件**

- 管理 IdM 或 用户管理员 角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。
- 现有权限。有关权限的详情，请参阅 在 IdM Web UI 中管理权限。

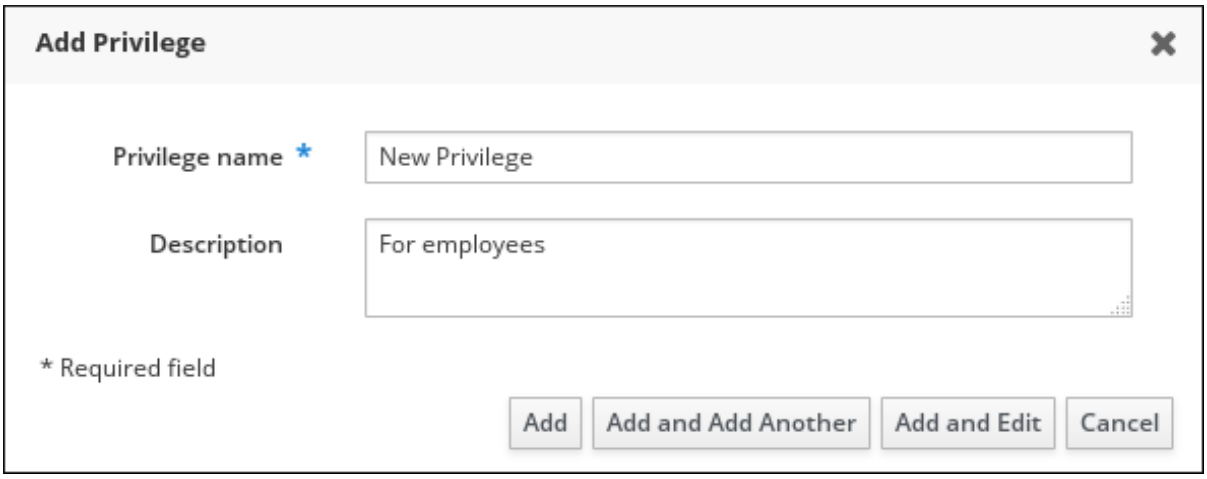
**流程**

1. 要添加一个新特权，请在 IPA Server 选项卡中打开 Role-Based Access Control 子菜单，然后选择 Privileges ：




2. 此时会打开权限列表。点击特权列表顶部的 Add 按钮：

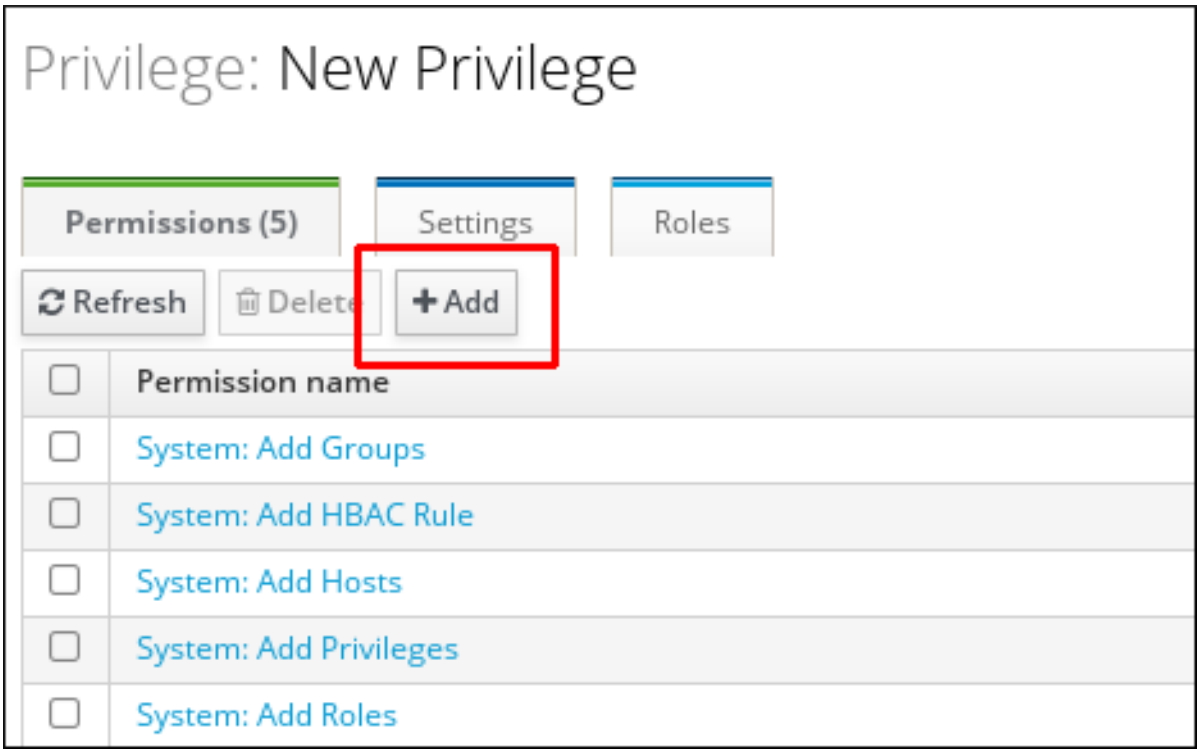
3. 此时会打开 **Add Privilege** 表单。输入特权的名称和描述：



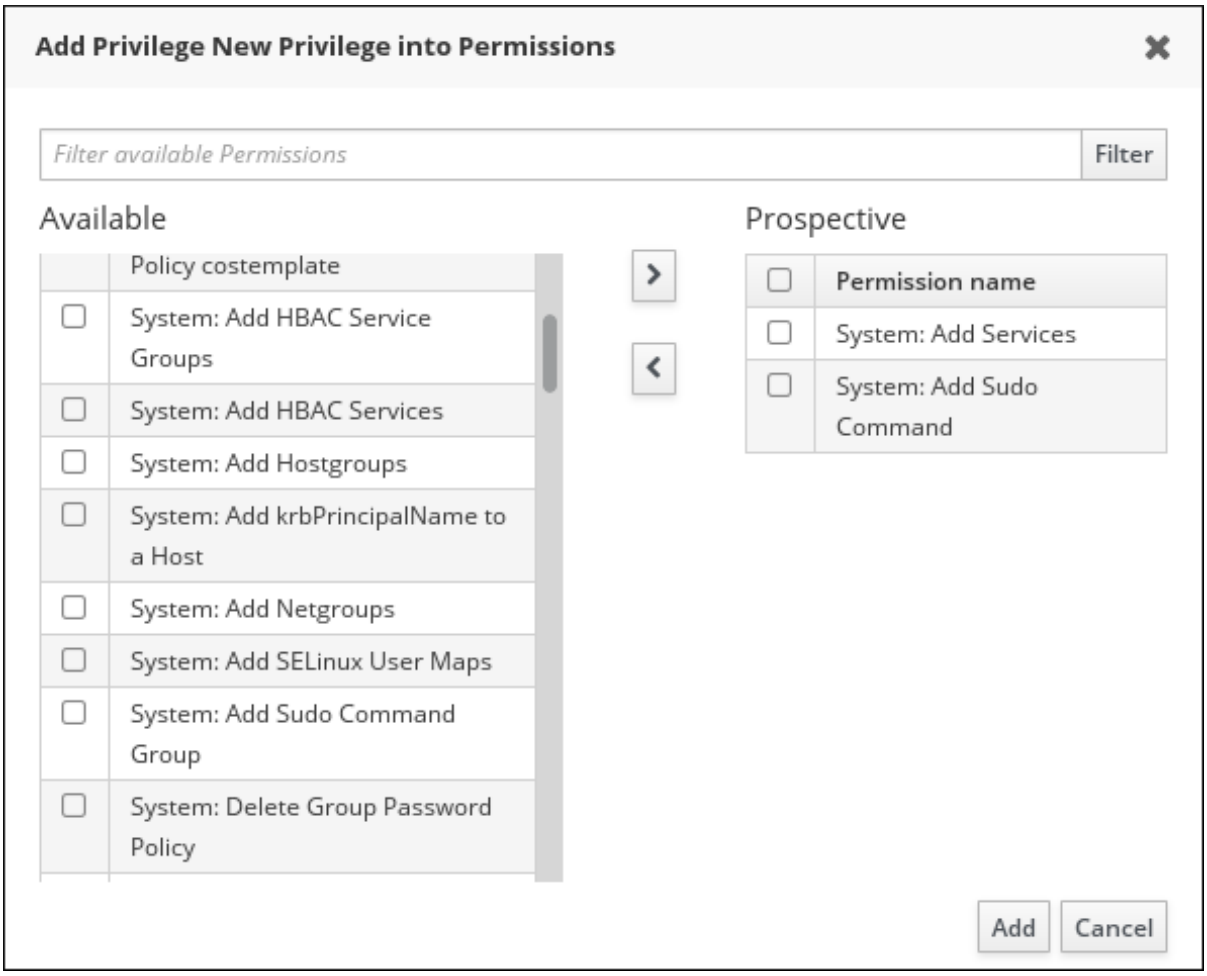

4. 单击 **Add and Edit** 按钮，以保存新特权，并继续特权配置页面来添加权限。

5. 单击特权列表中的特权名称，来编辑特权属性。此时会打开特权配置页面。

6. **Permissions** 选项卡显示选定的特权中包含的权限列表。点击列表顶部的 **Add** 按钮向特权添加权限：



7. 勾选每个要添加权限的名称旁边的复选框，并使用 > 按钮将权限移到 **Prospective** 列中：




8. 单击 **Add** 按钮进行确认。

9. *可选。*如果您需要删除权限，请在相关权限旁勾选复选框后单击 **Delete** 按钮：**Remove privileges from permissions** 对话框将打开。

10. *可选。*如果您需要删除现有的特权，请在勾选列表中其名称旁边的复选框后单击 **Delete** 按钮：**Remove privileges** 对话框将打开。

## 28.4. 在 IDM WEB UI 中管理角色

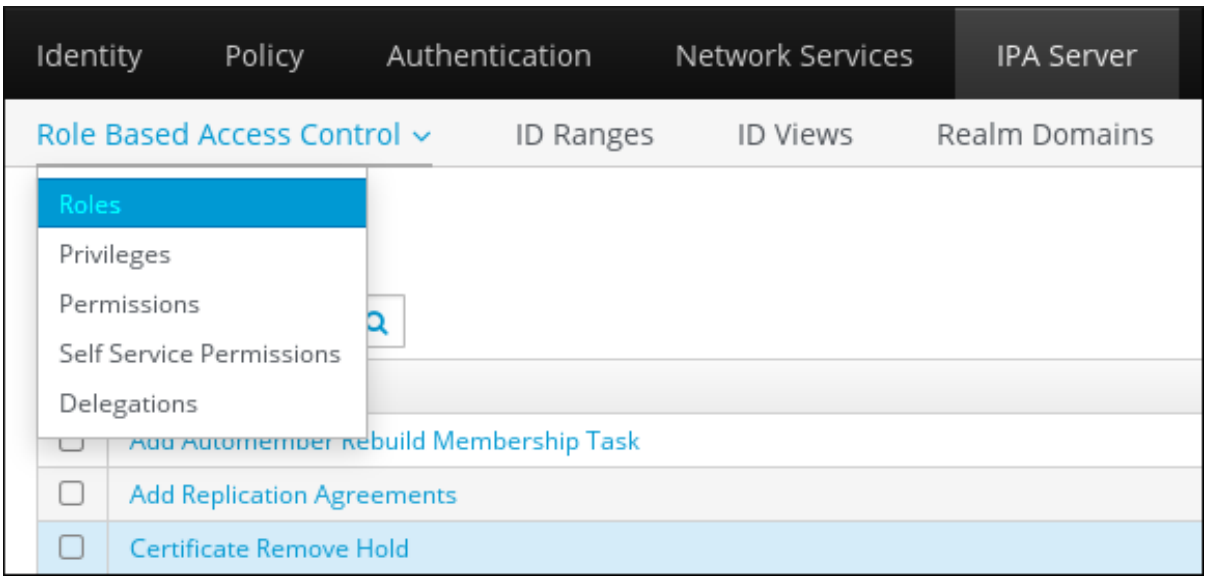
按照以下流程，使用 Web 界面(IdM Web UI)管理身份管理(IdM)中的角色。

**先决条件**

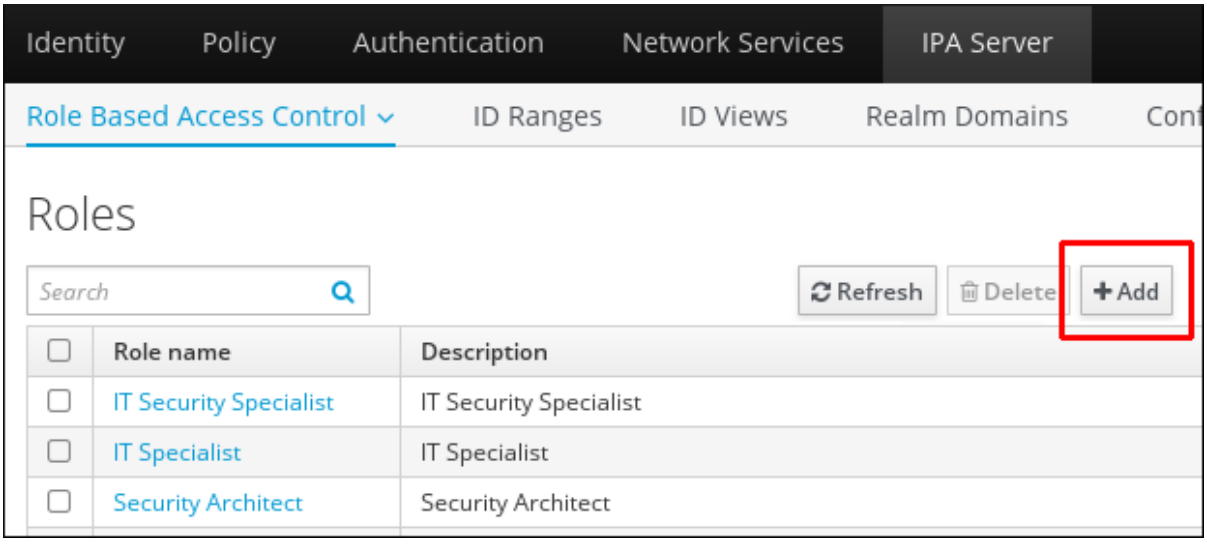
- 管理 IdM 或 用户管理员 角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。
- 现有特权。有关特权的详情，请参阅 在 IdM Web UI 中管理特权。

**流程**

1. 要添加一个新角色，请在 IPA Server 选项卡中打开 Role-Based Access Control 子菜单，然后选择 Role ：




2. 角色列表会打开。单击基于角色的访问控制指令列表顶部的 Add 按钮。

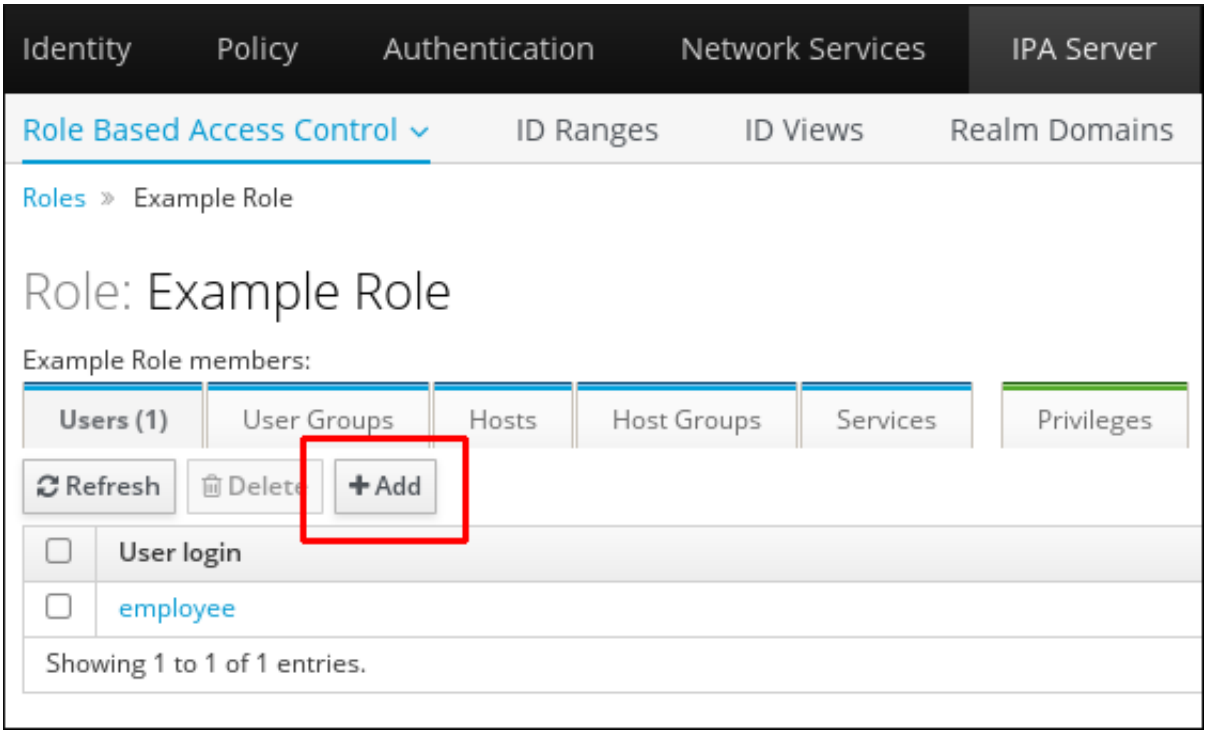



3. **此时会打开 Add Role 表单。输入角色名称和描述：**

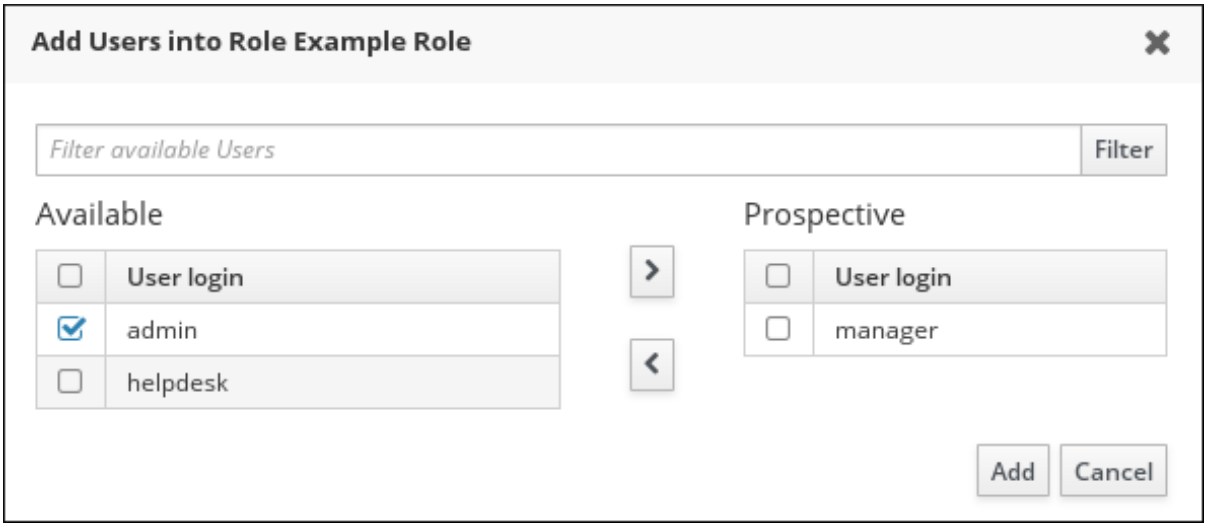

4. **单击 Add and Edit 按钮，来保存新角色，再前往角色配置页面来添加特权和用户。**

5. **单击角色列表中的角色名称，来编辑角色的属性。角色配置页面将打开。**

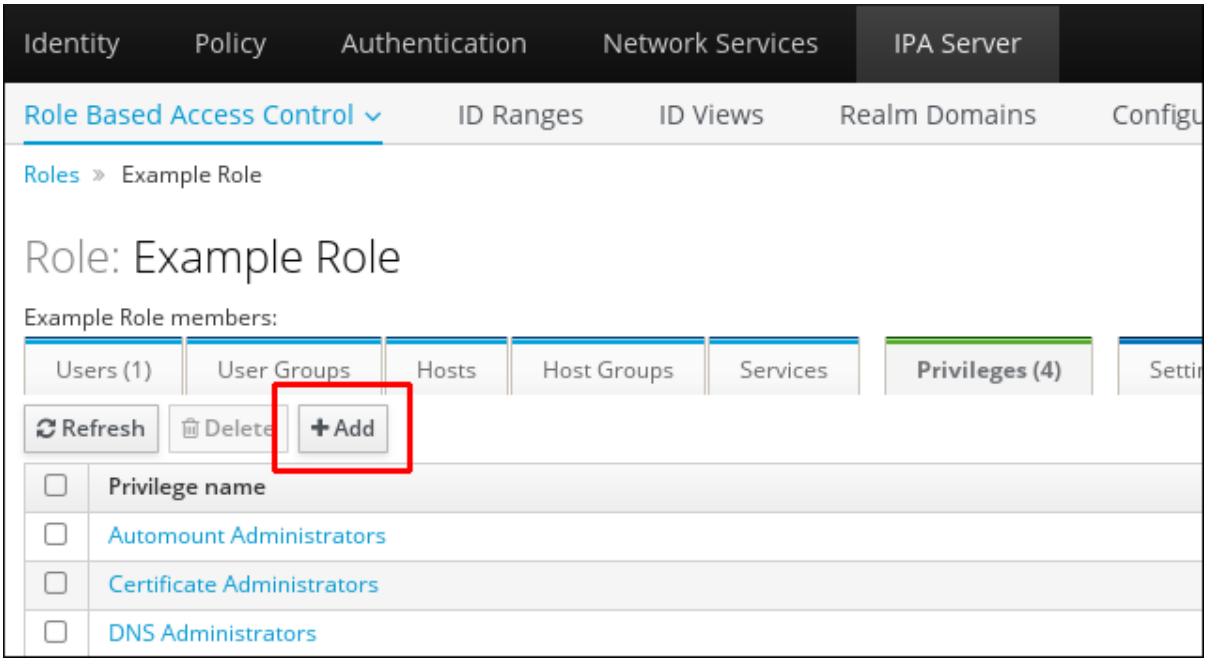
6. **单击相关列表顶部的 Add 按钮，使用 Users 、Users Groups、Hosts、Host Groups 或 Services 选项卡来添加成员。**



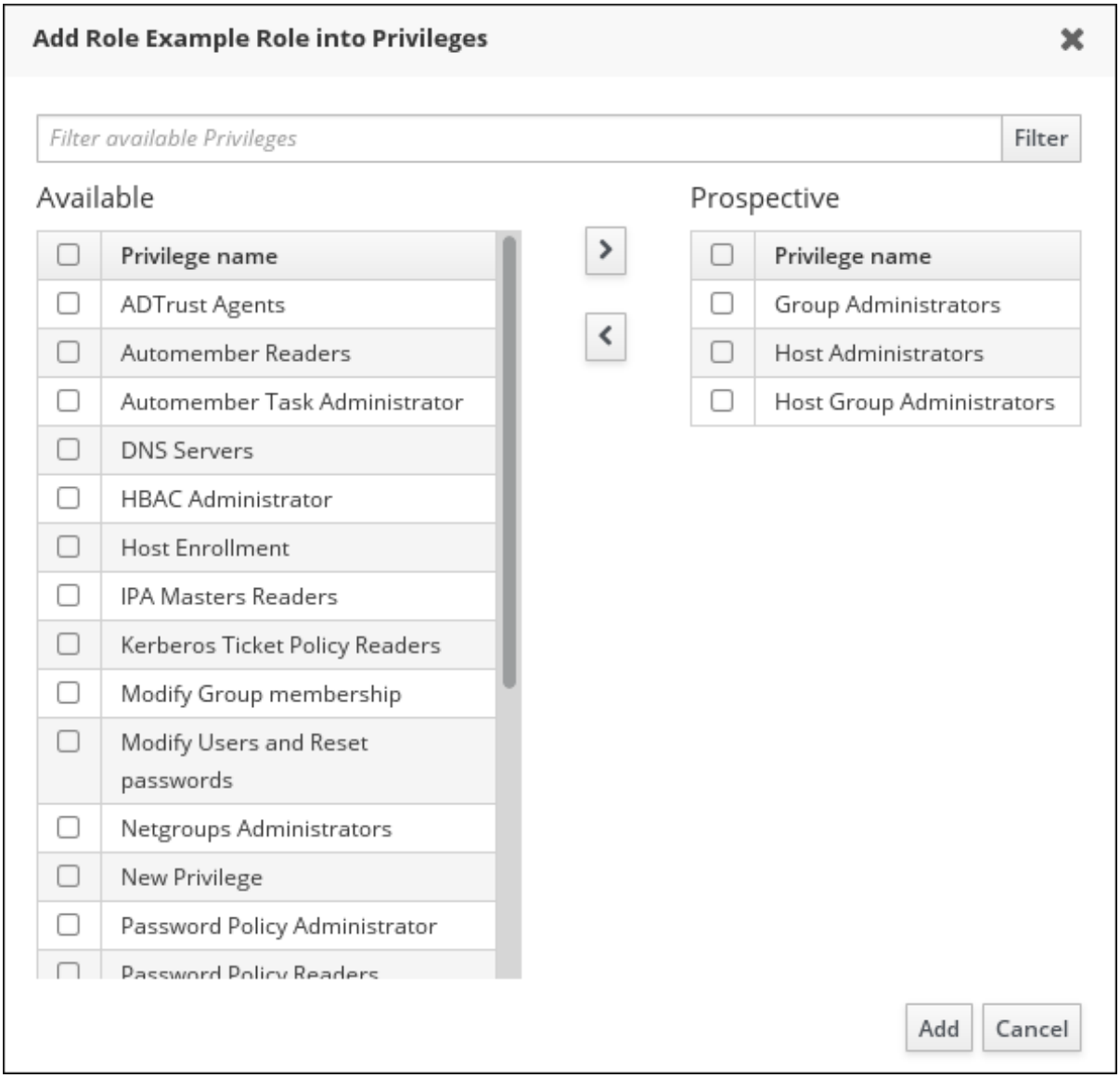

7. **在打开的窗口中，选择左侧的成员，并使用 > 按钮将它们移到 Prospective 列中。**




8. **在 Privileges 选项卡的顶部，单击 Add。**




9. **选择左侧的特权，并使用 > 按钮将它们移到 Prospective 列中。**




10. 单击 **Add** 按钮保存。

11. *可选。*如果您需要从角色中删除特权或成员，请在勾选您要删除的实体名称旁边的复选框后单击 **Delete** 按钮。此时会打开一个对话框。

12. *可选。*如果您需要删除现有角色，请在勾选列表中其名称旁边的复选框后单击 **Delete** 按钮，来显示 **Remove roles** 对话框。

# 第 29 章 在 IDM 中使用 ANSIBLE PLAYBOOK 管理基于角色的访问控制

基于角色的访问控制 (RBAC) 是一种基于角色和特权定义的策略中立访问控制机制。在 Identity Management (IdM) 中的 RBAC 组件是角色、权限和权限：

- Permissions 授予执行特定任务的权利，如添加或删除用户、修改组和启用读权限。
- Privileges（特权） 结合了权限，例如添加新用户所需的所有权限。
- Roles（角色） 向用户、用户组、主机或主机组授予一组特权。

尤其是在大型公司，使用 RBAC 可以帮助创建具有各个职责领域的管理员分层系统。

本章介绍了使用 Ansible playbook 管理 RBAC 时执行的以下操作：

- IdM 中的权限
- 默认管理的权限
- IdM 中的特权
- IdM 中的角色
- IdM 中的预定义角色
- 使用 Ansible 确保存在带有特权的 IdM RBAC 角色
- 使用 Ansible 确保缺少 IdM RBAC 角色

- 使用 Ansible 确保为一组用户分配 IdM RBAC 角色
- 使用 Ansible 确保没有将特定用户分配给 IdM RBAC 角色
- 使用 Ansible 确保服务是 IdM RBAC 角色的成员
- 使用 Ansible 确保主机是 IdM RBAC 角色的成员
- 使用 Ansible 确保主机组是 IdM RBAC 角色的成员

## 29.1. IDM 中的权限

权限是基于角色的访问控制的最低级别单元，它们定义这些操作所应用到的 LDAP 条目。与构建块类似，可以根据需要将权限分配给多个特权。
一个或多个权利定义了允许的操作 ：

- write
- 读取
- search
- compare
- add
- delete
- all

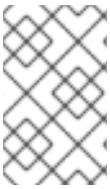
这些操作适用于三个基本目标：

- subtree：域名 (DN)；此 DN 下的子树
- target filter：LDAP 过滤器
- target：可以带有通配符的 DN 指定条目

此外，以下方便选项可设置对应的属性：

- type：对象类型（用户、组等）；设置subtree 和 target filter
- memberof：组成员；设置 target filter
- targetgroup：授予修改特定组的权限（如授予管理组成员资格的权限）；设置 target

使用 IdM 权限，您可以控制哪些用户有权访问哪些对象，甚至控制这些对象的属性。IdM 允许您允许或阻止单个属性，或更改特定 IdM 功能（如用户、组或 sudo）的所有可见性，适用于所有匿名用户、所有经过身份验证的用户，或者只更改一组特定的特权用户。
例如，如果管理员只想将用户或组的访问权限限制到这些用户或组需要访问的特定部分，并且使其他部分完全隐藏于他们，此方法的灵活性对管理员很有用。



注意

权限不能包含其他权限。

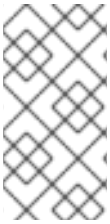
## 29.2. 默认管理的权限

管理的权限是 IdM 默认附带的权限。它们的行为与用户创建的其他权限类似，但有以下区别：

- 您无法删除它们或修改其名称、位置和目标属性。

- 它们有三组属性：
  - Default 属性，用户无法修改它们，因为它们由 IdM 管理
  - Included 属性，它们是用户添加的额外属性
  - Excluded 属性，这些属性由用户删除

管理的权限适用于 default 和 included 属性集中显示的所有属性，但不应用到排除集中的所有属性。



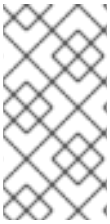
注意

虽然您无法删除受管权限，但将其绑定类型设置为权限并从所有特权中删除托管权限会有效地禁用该权限。

所有受管权限的名称都以 System: 开头，例如 System: Add Sudo rule 或 System: Modify Services。IdM 的早期版本将不同的方案用于默认权限。例如，用户无法删除它们，而只能将它们分配到特权。这些默认权限大部分已转换为受管权限，但以下权限仍使用以前的方案：

- 添加自动成员重新构建成员身份任务
- 添加配置子条目
- 添加复制协议
- 证书删除冻结
- 从 CA 获取证书状态
- 读取 DNA 范围

- 修改 DNA 范围
- 读取 PassSync Manager 配置
- 修改 PassSync Manager 配置
- 阅读复制协议
- 修改复制协议
- 删除复制协议
- 读取 LDBM 数据库配置
- 请求证书
- 请求证书忽略 CA ACL
- 从不同主机请求证书
- 从 CA 检索证书
- 吊销证书
- 写入 IPA 配置



**注意**

如果您试图通过命令行修改受管权限，系统不允许更改您无法修改的属性，命令会失败。如果您试图从 Web UI 修改受管权限，则无法修改的属性将被禁用。

## 29.3. IDM 中的特权

特权是一组适用于角色的权限。
虽然权限提供了执行单个操作的权限，但某些 IdM 任务需要多个权限才能成功。因此，特权组合了执行特定任务所需的不同权限。
例如，为新 IdM 用户设置帐户需要以下权限：

- 创建新用户条目
- 重置用户密码
- 将新用户添加到默认 IPA 用户组

将这三个低级别任务合并到一个更高级别的任务中，例如名为 Add User，可使系统管理员更加轻松地管理角色。IdM 已包含几个默认权限。除了用户和用户组外，还将特权分配到主机和主机组，以及网络服务。这种方法允许精细控制一组主机上使用特定网络服务的操作。

**注意**

特权可能不包含其他特权。

## 29.4. IDM 中的角色

角色是用户为角色指定的特权列表。
实际上，权限授予执行给定低级别任务（如创建用户条目和向组添加一个条目）的能力，特权将高级别任务所需的一个或多个这些权限（如在给定组中创建新用户）组合在一起。角色根据需要收集权限：例如，用户管理员角色能够添加、修改和删除用户。

**重要**

角色用于对允许的操作进行分类。它们不用作实施特权升级或防止特权升级的工具。

**注意**

**角色不能包含其他角色。**

## 29.5. IDENTITY MANAGEMENT 中的预定义角色

**Red Hat Identity Management 提供以下预定义角色范围：**

**表 29.1. 身份管理中的预定义角色**

| 角色 | 特权 | 描述 |
| --- | --- | --- |
| Enrollment Administrator | 主机注册 | 负责客户端或主机、注册 |
| helpdesk | 改用户和重置密码，修改组成员身份 | 负责执行简单的用户管理任务 |
| IT Security Specialist | Netgroups 管理员, HBAC 管理员, Sudo 管理员 | 负责管理安全策略，如基于主机的访问控制、sudo 规则 |
| IT Specialist | 主机管理员、主机组管理员、服务管理员、自动装载管理员 | 负责管理主机 |
| Security Architect | 委派管理员、复制管理员、写 IPA 配置、密码策略管理员 | 负责管理身份管理环境、创建信任、创建复制协议 |
| User Administrator | 用户管理员、组管理员、阶段用户管理员 | 负责创建用户和组 |

## 29.6. 使用 ANSIBLE 确保存在带有特权的 IDM RBAC 角色

**要对身份管理 (IdM) 中的资源 (IdM) 中的资源进行更加精细的控制，请创建自定义角色。**

**以下流程描述了如何使用 Ansible playbook 为新的 IdM 自定义角色定义特权并确保其存在。在这个示例中，新的 user_and_host_administrator 角色默认包含 IdM 中的以下权限的唯一组合：**

- **Group Administrators**
- **User Administrators**

- Stage User Administrators
- Group Administrators

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*<MyPlaybooks>*/ 目录：

```
$ cd ~/<MyPlaybooks>/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/role/ 目录的 role-member-user-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-user-present.yml role-member-user-present-copy.yml
```

3. 打开 role-member-user-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 iparole 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为新角色的名称。
- 将 privilege 列表设置为您要包含在新角色中的 IdM 权限的名称。
- （可选）将 user 变量设置为您要授予新角色的用户名称。
- （可选）将 group 变量设置为要授予新角色的组的名称。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: user_and_host_administrator
      user: idm_user01
      group: idm_group01
      privilege:
      - Group Administrators
      - User Administrators
      - Stage User Administrators
      - Group Administrators
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-user-present-copy.yml
```

其它资源

- 请参阅 使用 Ansible Vault 加密内容。
- 请参阅 IdM 中的角色。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-role 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/iparole 目录中的 playbook 示例。

## 29.7. 使用 ANSIBLE 确保缺少 IDM RBAC 角色

作为管理身份管理 (IdM) 中基于角色的访问控制 (RBAC) 的系统管理员，您可能希望确保没有过时的角色，以便任何管理员不会意外将它分配给任何用户。

以下流程描述了如何使用 Ansible playbook 来确保缺少角色。以下示例描述了如何确保 IdM 中不存在自定义 user_and_host_administrator 角色。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。

- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*<MyPlaybooks>*/ 目录：

```
$ cd ~/<MyPlaybooks>/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/role/ 目录的 role-is-absent.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-is-absent.yml role-is-absent-copy.yml
```

3. 打开 role-is-absent-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 iparole 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为角色的名称。
- 确保 state 变量设置为 absent。

这是当前示例修改的 Ansible playbook 文件：

```
---
```

```
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: user_and_host_administrator
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-is-absent-copy.yml
```

**其它资源**

- 请参阅 使用 Ansible Vault 加密内容。
- 请参阅 IdM 中的角色。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-role Markdown 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/iparole 目录中的 playbook 示例。

## 29.8. 使用 ANSIBLE 确保为一组用户分配 IDM RBAC 角色

作为管理身份管理 (IdM) 中基于角色的访问控制 (RBAC) 的系统管理员，您可能希望为一组特定的用户（如初级管理员）分配角色。

以下示例描述了如何使用 Ansible playbook 来确保为 junior_sysadmins 分配内置 IdM RBAC helpdesk 角色。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*<MyPlaybooks>*/ 目录：

```
$ cd ~/<MyPlaybooks>/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/role/ 目录的 role-member-group-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-group-present.yml
role-member-group-present-copy.yml
```

3. 打开 role-member-group-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 iparole 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为您要分配的角色的名称。
- 将 group 变量设置为组的名称。
- 将 action 变量设置为 member。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: helpdesk
      group: junior_sysadmins
      action: member
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-group-present-copy.yml
```

**其它资源**

- 请参阅 使用 Ansible Vault 加密内容。

- 请参阅 IdM 中的角色。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-role Markdown 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/iparole 目录中的 playbook 示例。

## 29.9. 使用 ANSIBLE 确保没有将特定用户分配给 IDM RBAC 角色

作为系统管理员，在身份管理 (IdM) 中管理基于角色的访问控制 (RBAC)，您可能需要确保在特定用户已移至公司内的不同位置后，不会为其分配 RBAC 角色。

以下流程描述了如何使用 Ansible playbook 来确保没有将名为 user_01 和 user_02 的用户分配到 helpdesk 角色。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*<MyPlaybooks>*/ 目录：

```
$ cd ~/<MyPlaybooks>/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/role/ 目录的 role-member-user-absent.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-user-absent.yml role-member-user-absent-copy.yml
```

3. 打开 role-member-user-absent-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 iparole 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为您要分配的角色的名称。
- 将 user 列表设置为用户的名称。
- 将 action 变量设置为 member。
- 将 state 变量设置为 absent。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
```

```
tasks:
- iparole:
   ipaadmin_password: "{{ ipaadmin_password }}"
   name: helpdesk
   user
   - user_01
   - user_02
   action: member
   state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-user-absent-copy.yml
```

**其它资源**

- 请参阅 使用 Ansible Vault 加密内容。
- 请参阅 IdM 中的角色。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-role Markdown 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/iparole 目录中的 playbook 示例。

## 29.10. 使用 ANSIBLE 确保服务是 IDM RBAC 角色的成员

作为管理身份管理 (IdM) 中基于角色的访问控制 (RBAC) 的系统管理员，您可能希望确保注册 IdM 的特定服务是特定角色的成员。以下示例描述了如何确保自定义 web_administrator 角色可以管理 client01.idm.example.com 服务器上运行的 HTTP 服务。

**先决条件**

- 您知道 IdM 管理员密码。

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- web_administrator 角色存在于 IdM 中。
- IdM 中存在 HTTP/client01.idm.example.com@IDM.EXAMPLE.COM 服务。

**流程**

1. 进入 ~/*<MyPlaybooks>*/ 目录：

```
$ cd ~/<MyPlaybooks>/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/role/ 目录的 role-member-service-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-service-present-absent.yml role-member-service-present-copy.yml
```

3. 打开 role-member-service-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 iparole 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为您要分配的角色的名称。
- 将 service 列表设置为服务的名称。
- 将 action 变量设置为 member。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web_administrator
      service:
      - HTTP/client01.idm.example.com
      action: member
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-service-present-copy.yml
```

**其它资源**

- 请参阅 使用 Ansible Vault 加密内容。
- 请参阅 IdM 中的角色。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-role Markdown 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/iparole 目录中的 playbook 示例。

## 29.11. 使用 ANSIBLE 确保主机是 IDM RBAC 角色的成员

作为在身份管理 (IdM) 中管理基于角色的访问控制的系统管理员，您可能希望确保特定的主机或主机组与特定角色关联。以下示例描述了如何确保自定义 web_administrator 角色可以管理运行 HTTP 服务的 client01.idm.example.com IdM 主机。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- web_administrator 角色存在于 IdM 中。
- client01.idm.example.com 主机存在于 IdM 中。

**流程**

1. 进入 ~/*<MyPlaybooks>*/ 目录：

```
$ cd ~/<MyPlaybooks>/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/role/ 目录的 role-member-host-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-host-present.yml role-member-host-present-copy.yml
```

3. 打开 role-member-host-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 iparole 任务部分设置以下变量来调整文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
   - 将 name 变量设置为您要分配的角色的名称。
   - 将 host 列表设置为主机的名称。

   这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  gather_facts: no

  vars_files:
```

```
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web_administrator
      host:
      - client01.idm.example.com
      action: member
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-host-present-copy.yml
```

**其它资源**

- 请参阅 使用 Ansible Vault 加密内容。
- 请参阅 IdM 中的角色。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-role Markdown 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/iparole 目录中的 playbook 示例。

## 29.12. 使用 ANSIBLE 确保主机组是 IDM RBAC 角色的成员

作为在身份管理 (IdM) 中管理基于角色的访问控制的系统管理员，您可能希望确保特定的主机或主机组与特定角色关联。以下示例描述了如何确保自定义 web_administrator 角色可以管理运行 HTTP 服务的 IdM 主机组的 web_servers 组。

**先决条件**

- 您知道 IdM 管理员密码。

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- web_administrator 角色存在于 IdM 中。
- web_servers 主机组存在于 IdM 中。

**流程**

1. 进入 ~/*<MyPlaybooks>*/ 目录：

```
$ cd ~/<MyPlaybooks>/
```

2. 创建位于 /usr/share/doc/ansible-freeipa/playbooks/role/ 目录的 role-member-hostgroup-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-hostgroup-present.yml role-member-hostgroup-present-copy.yml
```

3. 打开 role-member-hostgroup-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 iparole 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为您要分配的角色的名称。
- 将 hostgroup 列表设置为 hostgroup 的名称。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web_administrator
      hostgroup:
      - web_servers
      action: member
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-hostgroup-present-copy.yml
```

**其它资源**

- 请参阅 使用 Ansible Vault 加密内容。

- 请参阅 IdM 中的角色。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-role Markdown 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/iparole 目录中的 playbook 示例。

# 第 30 章 使用 ANSIBLE PLAYBOOK 管理 RBAC 特权

基于角色的访问控制 (RBAC) 是一种基于角色、特权和权限定义的策略中立访问控制机制。尤其是在大型公司，使用 RBAC 可以帮助创建具有各个职责领域的管理员分层系统。

本章介绍了以下操作，以使用 Ansible playbook 管理身份管理 (IdM) 中的 RBAC 特权：

- 使用 Ansible 确保存在自定义 RBAC 特权
- 使用 Ansible 确保自定义 IdM RBAC 特权中存在成员权限
- 使用 Ansible 确保 IdM RBAC 特权不包括权限
- 使用 Ansible 重命名自定义 IdM RBAC 特权
- 使用 Ansible 确保缺少 IdM RBAC 特权

**先决条件**

- 您已了解 RBAC 的概念和原则。

## 30.1. 使用 ANSIBLE 确保存在自定义 IDM RBAC 特权

要在 Identity Management (IdM) 基于角色的访问控制 (RBAC) 中有一个完全设计的自定义权限，您需要逐步进行：

1. 创建没有附加权限的特权。
2. 将您选择的权限添加到特权。

以下流程描述了如何使用 Ansible playbook 创建空特权，以便稍后您可以向它添加权限。这个示例描述了如何创建名为 full_host_administration 的特权，它旨在组合与主机管理相关的所有 IdM 权限。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 生成位于 /usr/share/doc/ansible-freeipa/playbooks/privilege/ 目录中的 privilege-present.yml 文件副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-present.yml privilege-present-copy.yml
```

3. 打开 privilege-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipaprivilege 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为新特权 full_host_administration 的名称。
- （可选）利用 description 变量描述特权。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Privilege present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure privilege full_host_administration is present
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: full_host_administration
      description: This privilege combines all IdM permissions related to host
administration
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory privilege-
present-copy.yml
```

## 30.2. 使用 ANSIBLE 确保自定义 IDM RBAC 特权中存在成员权限

要在 Identity Management (IdM) 基于角色的访问控制 (RBAC) 中有一个完全设计的自定义权限，您需要逐步进行：

1. 创建没有附加权限的特权。

2. 将您选择的权限添加到特权。

以下流程描述了如何使用 Ansible playbook 向上一步中创建的特权添加权限。这个示例描述了如何将与主机管理相关的所有 IdM 权限添加到名为 full_host_administration 的特权中。默认情况下，权限在 Host Enrollment、Host Administrators 和 Host Group Administrator 特权之间分发。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- *full_host_administration* 特权存在。有关如何使用 Ansible 创建特权的详情，请参阅 使用 Ansible 确保自定义 IdM RBAC 特权存在。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 生成位于 /usr/share/doc/ansible-freeipa/playbooks/privilege/ 目录中的 privilege-member-present.yml 文件副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-member-present.yml
privilege-member-present-copy.yml
```

3. 打开 privilege-member-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipaprivilege 任务部分设置以下变量来调整文件：

- 调整任务的 name，使其与您的用例对应。
- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为特权的名称。
- 将 permission 列表设置为您要包含在权限中的权限名称。
- 确保 action 变量设置为 member。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Privilege member present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that permissions are present for the "full_host_administration"
privilege
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: full_host_administration
      permission:
```

```
    - "System: Add krbPrincipalName to a Host"
    - "System: Enroll a Host"
    - "System: Manage Host Certificates"
    - "System: Manage Host Enrollment Password"
    - "System: Manage Host Keytab"
    - "System: Manage Host Principals"
    - "Retrieve Certificates from the CA"
    - "Revoke Certificate"
    - "System: Add Hosts"
    - "System: Add krbPrincipalName to a Host"
    - "System: Enroll a Host"
    - "System: Manage Host Certificates"
    - "System: Manage Host Enrollment Password"
    - "System: Manage Host Keytab"
    - "System: Manage Host Keytab Permissions"
    - "System: Manage Host Principals"
    - "System: Manage Host SSH Public Keys"
    - "System: Manage Service Keytab"
    - "System: Manage Service Keytab Permissions"
    - "System: Modify Hosts"
    - "System: Remove Hosts"
    - "System: Add Hostgroups"
    - "System: Modify Hostgroup Membership"
    - "System: Modify Hostgroups"
    - "System: Remove Hostgroups"
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory privilege-member-present-copy.yml
```

### 30.3. 使用 ANSIBLE 确保 IDM RBAC 特权不包括权限

作为身份管理系统管理员 (IdM)，您可以自定义 IdM 基于角色的访问控制。

以下流程描述了如何使用 Ansible playbook 从特权中删除权限。示例描述了如何从默认 Certificate Administrators 特权中删除 Request Certificates ignoring CA ACLs 权限，例如，管理员认为它存在安全风险。

**先决条件**

- 您知道 IdM 管理员密码。

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 生成位于 /usr/share/doc/ansible-freeipa/playbooks/privilege/ 目录中的 privilege-member-present.yml 文件副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-member-absent.yml
privilege-member-absent-copy.yml
```

3. 打开 privilege-member-absent-copy.yml Ansible playbook 文件以进行编辑。
4. 通过在 ipaprivilege 任务部分设置以下变量来调整文件：
   - 调整任务的 name，使其与您的用例对应。

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为特权的名称。
- 将 permission 列表设置为您要从特权中删除的权限名称。
- 确保 action 变量设置为 member。
- 确保 state 变量设置为 absent。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Privilege absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "Request Certificate ignoring CA ACLs" permission is absent
from the "Certificate Administrators" privilege
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: Certificate Administrators
      permission:
      - "Request Certificate ignoring CA ACLs"
      action: member
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory privilege-
member-absent-copy.yml
```

## 30.4. 使用 ANSIBLE 重命名自定义 IDM RBAC 特权

作为身份管理系统管理员 (IdM)，您可以自定义 IdM 基于角色的访问控制。

以下流程描述了如何重命名权限，例如，您已从其中删除了一些权限。因此，特权的名称不再准确。在示例中，管理员将 full_host_administration 特权重命名为 limited_host_administration。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- *full_host_administration* 特权存在。有关如何添加特权的更多信息，请参阅 使用 Ansible 确保自定义 IdM RBAC 特权存在。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 生成位于 /usr/share/doc/ansible-freeipa/playbooks/privilege/ 目录中的 privilege-

present.yml 文件副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-present.yml rename-privilege.yml
```

3. 打开 rename-privilege.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipaprivilege 任务部分设置以下变量来调整文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
   - 将 name 变量设置为特权的当前名称。
   - 添加 rename 变量，并将它设置为特权的新名称。
   - 添加 state 变量，并将它设置为 重命名。

5. 重新命名 playbook 本身，例如：

```
---
- name: Rename a privilege
  hosts: ipaserver
```

6. 在 playbook 中重命名任务，例如：

```
[...]
tasks:
- name: Ensure the full_host_administration privilege is renamed to
limited_host_administration
  ipaprivilege:
  [...]
```

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Rename a privilege
```

```
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the full_host_administration privilege is renamed to
limited_host_administration
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: full_host_administration
      rename: limited_host_administration
      state: renamed
```

7. 保存该文件。

8. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory rename-
privilege.yml
```

### 30.5. 使用 ANSIBLE 确保缺少 IDM RBAC 特权

作为身份管理系统管理员 (IdM)，您可以自定义 IdM 基于角色的访问控制。以下流程描述了如何使用 Ansible playbook 来确保缺少 RBAC 特权。这个示例描述了如何确保缺少 CA administrator 特权。因此，admin 成为在 IdM 中管理证书颁发机构的唯一用户。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 *~/MyPlaybooks/* 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 生成位于 /usr/share/doc/ansible-freeipa/playbooks/privilege/ 目录中的 privilege-absent.yml 文件副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-absent.yml privilege-absent-copy.yml
```

3. 打开 privilege-absent-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipaprivilege 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。

- 将 name 变量设置为您要删除的特权的名称。

- 确保 state 变量设置为 absent。

5. 在 playbook 中重命名任务，例如：

```
[...]
tasks:
- name: Ensure privilege "CA administrator" is absent
  ipaprivilege:
  [...]
```

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Privilege absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure privilege "CA administrator" is absent
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: CA administrator
      state: absent
```

6. 保存该文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory privilege-absent-copy.yml
```

### 30.6. 其它资源

- 请参阅 IdM 中的特权。
- 请参阅 IdM 中的权限。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-privilege 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/ipaprivilege 目录中的 playbook 示例。

# 第 31 章 使用 ANSIBLE PLAYBOOK 在 IDM 中管理 RBAC 权限

基于角色的访问控制 (RBAC) 是一种基于角色、特权和权限定义的策略中立访问控制机制。尤其是在大型公司，使用 RBAC 可以帮助创建具有各个职责领域的管理员分层系统。

本章介绍了使用 Ansible playbook 管理身份管理 (IdM) 中 RBAC 权限时执行的以下操作：

- 使用 Ansible 确保存在 RBAC 权限
- 使用 Ansible 确保存在带有属性的 RBAC 权限
- 使用 Ansible 确保缺少 RBAC 权限
- 使用 Ansible 确保属性是 IdM RBAC 权限的成员
- 使用 Ansible 确保属性不是 IdM RBAC 权限的成员
- 使用 Ansible 重命名 IdM RBAC 权限

**先决条件**

- 您已了解 RBAC 的概念和原则。

## 31.1. 使用 ANSIBLE 确保存在 RBAC 权限

作为身份管理系统管理员 (IdM)，您可以自定义 IdM 基于角色的访问控制 (RBAC)。

以下流程描述了如何使用 Ansible playbook 确保 IdM 中存在权限，以便它可以添加到特权中。这个示例描述了如何确保以下目标状态：

- MyPermission 权限存在。

- MyPermission 权限只能应用到主机。
- 授予了包含权限的用户可以对条目执行以下所有可能的操作：
  - 写
  - 读
  - 搜索
  - 比较
  - 添加
  - 删除

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/permission/ 目录中的 permission-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-present.yml
permission-present-copy.yml
```

3. 打开 permission-present-copy.yml Ansible playbook 文件进行编辑。

4. 通过在 ipapermission 任务部分中设置以下变量来调整文件：

- 调整任务的 name，使其与您的用例对应。
- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为权限的名称。
- 将 object_type 变量设置为 host。
- 将 right 变量设置为 all。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Permission present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "MyPermission" permission is present
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
      object_type: host
      right: all
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-present-copy.yml
```

## 31.2. 使用 ANSIBLE 确保存在带有属性的 RBAC 权限

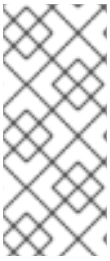
作为身份管理系统管理员 (IdM)，您可以自定义 IdM 基于角色的访问控制 (RBAC)。

以下流程描述了如何使用 Ansible playbook 确保 IdM 中存在权限，以便它可以添加到特权中。这个示例描述了如何确保以下目标状态：

- MyPermission 权限存在。
- MyPermission 权限只能用于添加主机。
- 获得了包含权限的用户可以在主机条目上执行以下所有可能的操作：
  - 写

- 读
- 搜索
- 比较
- 添加
- 删除

- 被授予特权的用户创建的主机条目包含 MyPermission 权限，可以具有 description 值。



**注意**

创建或修改权限时可以指定的属性类型不受 IdM LDAP 模式的限制。但是，当 object_type 是 host 时指定 attrs: car_licence，会导致在使用权限并为一个主机添加特定的 car 许可证时出现 ipa: ERROR: attribute "car-license" not allowed 错误。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/permission/ 目录中的 permission-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-present.yml
permission-present-with-attribute.yml
```

3. 打开 permission-present-with-attribute.yml Ansible playbook 文件进行编辑。

4. 通过在 ipapermission 任务部分中设置以下变量来调整文件：

- 调整任务的 name，使其与您的用例对应。
- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为权限的名称。
- 将 object_type 变量设置为 host。
- 将 right 变量设置为 all。
- 将 attrs 变量设置为 description。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Permission present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "MyPermission" permission is present with an attribute
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
      object_type: host
      right: all
      attrs: description
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-present-with-attribute.yml
```

**其它资源**

- 请参阅 RHEL 7 中的 *Linux 域身份、身份验证和策略指南* 中的 用户和组模式。

## 31.3. 使用 ANSIBLE 确保缺少 RBAC 权限

作为身份管理系统管理员 (IdM)，您可以自定义 IdM 基于角色的访问控制 (RBAC)。

以下流程描述了如何使用 Ansible playbook 确保 IdM 中缺少权限，因此无法将其添加到特权中。

**先决条件**

- 您知道 IdM 管理员密码。

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/permission/ 目录中的 permission-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-absent.yml
permission-absent-copy.yml
```

3. 打开 permission-absent-copy.yml Ansible playbook 文件进行编辑。

4. 通过在 ipapermission 任务部分中设置以下变量来调整文件：
   - 调整任务的 name，使其与您的用例对应。

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。

- 将 name 变量设置为权限的名称。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Permission absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "MyPermission" permission is absent
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-absent-copy.yml
```

## 31.4. 使用 ANSIBLE 确保属性是 IDM RBAC 权限的成员

作为身份管理系统管理员 (IdM)，您可以自定义 IdM 基于角色的访问控制 (RBAC)。

以下流程描述了如何使用 Ansible playbook 确保属性是 IdM 中 RBAC 权限的成员。因此，拥有权限的用户可以创建具有属性的条目。

示例描述了如何确保特权包含 MyPermission 权限的用户创建的主机条目可以具有 gecos 和 description 值。

**注意**

创建或修改权限时可以指定的属性类型不受 IdM LDAP 模式的限制。但是，当 object_type 是 host 时指定 attrs: car_licence，会导致在使用权限并为一个主机添加特定的 car 许可证时出现 ipa: ERROR: attribute "car-license" not allowed 错误。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- MyPermission 权限存在。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/permission/ 目录中的 permission-member-present.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-member-present.yml permission-member-present-copy.yml
```

3. 打开 permission-member-present-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipapermission 任务部分中设置以下变量来调整文件：

- 调整任务的 name，使其与您的用例对应。
- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为权限的名称。
- 将 attrs 列表设置为 description 和 gecos 变量。
- 确保 action 变量设置为 member。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Permission member present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "gecos" and "description" attributes are present in "MyPermission"
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
      attrs:
      - description
      - gecos
      action: member
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-member-present-copy.yml
```

## 31.5. 使用 ANSIBLE 确保属性不是 IDM RBAC 权限的成员

作为身份管理系统管理员 (IdM)，您可以自定义 IdM 基于角色的访问控制 (RBAC)。

以下流程描述了如何使用 Ansible playbook 确保属性不是 IdM 中 RBAC 权限的成员。因此，当拥有权限的用户在 IdM LDAP 中创建条目时，该条目不能具有与属性关联的值。

这个示例描述了如何确保以下目标状态：

- MyPermission 权限存在。
- 具有特权的用户创建的主机条目包含 MyPermission 权限，不能具有 description 属性。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- MyPermission 权限存在。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/permission/ 目录中的 permission-member-absent.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-member-absent.yml permission-member-absent-copy.yml
```

3. 打开 permission-member-absent-copy.yml Ansible playbook 文件进行编辑。

4. 通过在 ipapermission 任务部分中设置以下变量来调整文件：

- 调整任务的 name，使其与您的用例对应。

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。

- 将 name 变量设置为权限的名称。

- 将 attrs 变量设置为 description。

- 将 action 变量设置为 member。

- 确保 state 变量设置为 absent

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Permission absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that an attribute is not a member of "MyPermission"
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
      attrs: description
      action: member
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-member-absent-copy.yml
```

## 31.6. 使用 ANSIBLE 重命名 IDM RBAC 权限

作为身份管理系统管理员 (IdM)，您可以自定义 IdM 基于角色的访问控制。

以下流程描述了如何使用 Ansible playbook 重新命名权限。这个示例描述了如何将 MyPermission 重命名为 MyNewPermission。

**先决条件**

- 您知道 IdM 管理员密码。

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- MyPermission 存在于 IdM 中。
- IdM 中不存在 MyNewPermission。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作位于 /usr/share/doc/ansible-freeipa/playbooks/permission/ 目录中的 permission-renamed.yml 文件的副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-renamed.yml
permission-renamed-copy.yml
```

3. 打开 permission-renamed-copy.yml Ansible playbook 文件进行编辑。

4. 通过在 ipapermission 任务部分中设置以下变量来调整文件：

- 调整任务的 name，使其与您的用例对应。
- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为权限的名称。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Permission present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Rename the "MyPermission" permission
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
      rename: MyNewPermission
      state: renamed
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-renamed-copy.yml
```

## 31.7. 其它资源

- 请参阅 IdM 中的权限。
- 请参阅 IdM 中的特权。

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-permission 文件。

- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/ipapermission 目录中的 playbook 示例。

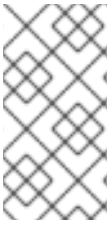
# 第 32 章 在 IDM 中管理用户密码

## 32.1. 谁可以更改 IDM 用户密码以及如何去做

没有权限更改其他用户密码的普通用户只能更改他们自己的个人密码。新密码必须满足适用于用户所属的组的 IdM 密码策略。有关配置密码策略的详情，请参考 定义 IdM 密码策略。

具有密码更改权限的管理员和用户可为新用户设置初始密码，并为现有用户重置密码。这些密码：

- 不必满足 IdM 密码策略。
- 在第一次成功登录后过期。当发生这种情况时，IdM 会提示用户立即更改过期的密码。要禁用此行为，请参阅 在 IdM 中启用密码重置，而不会在下一次登录时提示用户更改密码。



**注意**

LDAP 目录管理器(DM)用户可以使用 LDAP 工具更改用户密码。新密码可覆盖任何 IdM 密码策略。DM 设置的密码不会在第一次登录后过期。

## 32.2. 在 IDM WEB UI 中更改用户密码

作为身份管理(IdM)用户，您可以在 IdM Web UI 中更改用户密码。

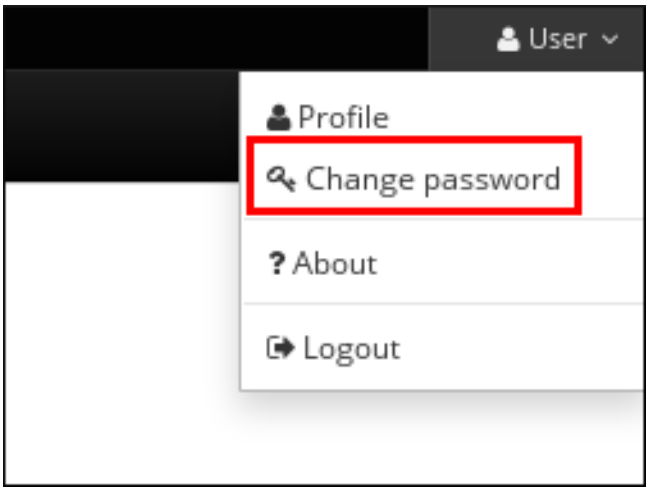
**先决条件**

- 已登陆到 IdM Web UI。

**流程**

1. 在右上角，点击 User name → Change password。

图 32.1. 重置密码




2. 输入当前的密码以及新密码。

## 32.3. 在 IDM WEB UI 中重置另一个用户的密码

作为身份管理(IdM)的管理员用户，您可以在 IdM Web UI 中更改其他用户的密码。

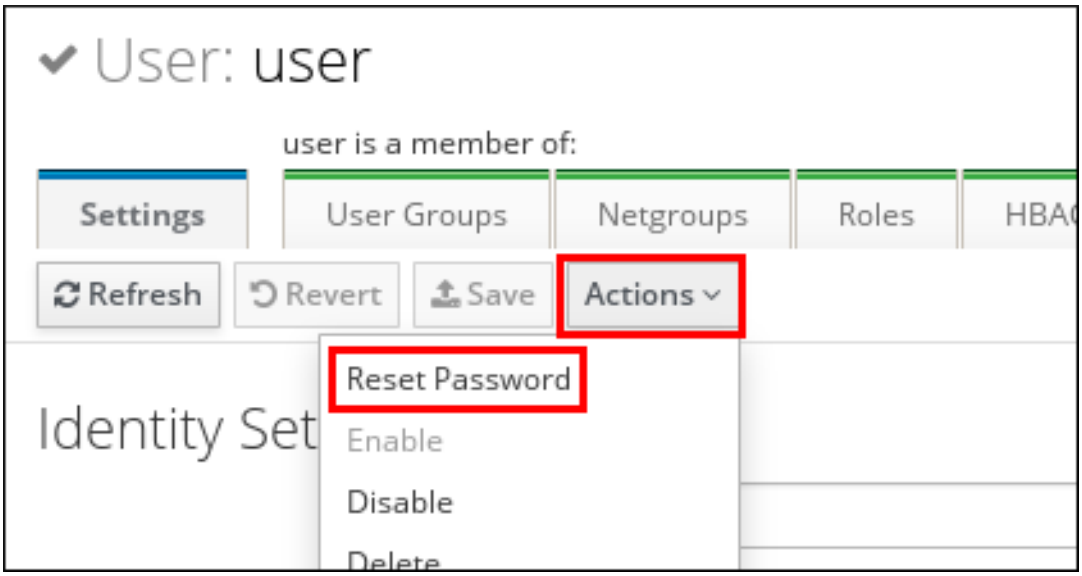
**先决条件**

- 您以管理员用户身份登录到 IdM Web UI。

**流程**

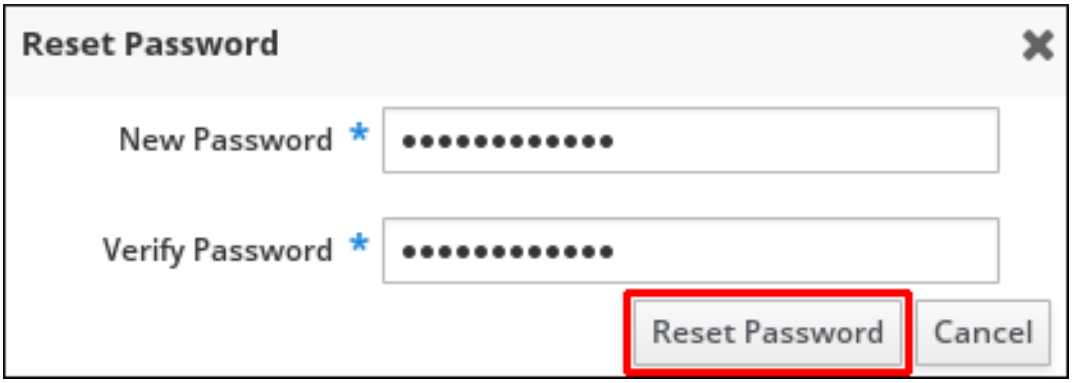
1. 选择 **Identity** → **Users**。
2. 单击要编辑的用户的名称。
3. 单击 **Actions** → **Reset password**。

图 32.2. 重置密码




4. 输入新密码，然后单击 **Reset Password**。

图 32.3. 确认新密码




## 32.4. 重置目录管理器用户密码

如果您丢失了身份管理(IdM)目录管理器密码，您可以重置它。

**先决条件**

- 您有 IdM 服务器的 root 访问权限。

**流程**

1. 使用 pwdhash 命令生成新的密码哈希。例如：

```
# pwdhash -D /etc/dirsrv/slapd-IDM-EXAMPLE-COM password
{PBKDF2_SHA256}AAAgABU0bKhyjY53NcxY33ueoPjOUWtI4iyYN5uW...
```

通过指定目录服务器配置的路径，您可以自动使用 nsslapd-rootpwstoragescheme 属性中设置的密码存储模式来加密新密码。

2. 在拓扑中的每个 IdM 服务器上执行以下步骤：

   a. 停止服务器上安装的所有 IdM 服务：

   ```
   # ipactl stop
   ```

   b. 编辑 /etc/dirsrv/IDM-EXAMPLE-COM/dse.ldif 文件，并将 nsslapd-rootpw 属性设为 pwdhash 命令所生成的值：

   ```
   nsslapd-rootpw:
   {PBKDF2_SHA256}AAAgABU0bKhyjY53NcxY33ueoPjOUWtI4iyYN5uW...
   ```

   c. 启动服务器上安装的所有 IdM 服务：

```
# ipactl start
```

## 32.5. 在 IDM CLI 中更改您的用户密码或重置另一个用户的密码

您可以使用身份管理(IdM)命令行界面(CLI)更改用户密码。如果您是管理用户，您可以使用 CLI 重置另一个用户的密码。

### 先决条件

- 您已获得了 IdM 用户的票据授予票(TGT)。
- 如果要重置另一个用户的密码，您必须获得 IdM 中管理用户的 TGT。

### 流程

- 输入 ipa user-mod 命令，以及用户名和 --password 选项。命令将提示您输入新密码。

```
$ ipa user-mod idm_user --password
Password:
Enter Password again to verify:
--------------------
Modified user "idm_user"
--------------------
...
```

注意

您还可以使用 ipa passwd *idm_user* 命令，而不是 ipa user-mod。

## 32.6. 在 IDM 中启用密码重置，而不会在下一次登录时提示用户更改密码

默认情况下，当管理员重置了另一个用户的密码后，密码会在第一次成功登录后过期。作为 IdM 目录管理者，您可以为单个的 IdM 管理员指定以下权限：

- 它们可以执行密码更改操作，而无需用户在第一次登录时更改其密码。
- 它们可以绕过密码策略，从而不会应用强度或历史记录强制。

警告

绕过密码策略可能会构成安全威胁。当您选择要授予这些额外特权的用户时要谨慎。

先决条件

- 您知道目录管理者密码。

流程

1. 在域中的每个身份管理(IdM)服务器上进行以下更改：

   a. 输入 ldapmodify 命令来修改 LDAP 条目。指定 IdM 服务器的名称和 389 端口，然后按回车：

   ```
   $ ldapmodify -x -D "cn=Directory Manager" -W -h server.idm.example.com -p 389
   Enter LDAP Password:
   ```

   b. 输入 Directory Manager 密码。

   c. 输入 ipa_pwd_extop 密码同步条目的可区别的名称，然后按回车

   ```
   dn: cn=ipa_pwd_extop,cn=plugins,cn=config
   ```

   d. 指定更改的 modify 类型，并按回车：

   ```
   changetype: modify
   ```

   e. 指定您希望 LDAP 执行哪种类型的修改，以及指定对哪个属性的修改。按回车：

   ```
   add: passSyncManagersDNs
   ```

   f. 在 passSyncManagersDNs 属性中指定管理用户帐户。属性是多值的。例如，要授予 admin 用户目录管理者重置密码的权力：

   ```
   passSyncManagersDNs: \
   uid=admin,cn=users,cn=accounts,dc=example,dc=com
   ```

   g. 按回车两次以停止编辑条目。

整个过程如下所示：

```
$ ldapmodify -x -D "cn=Directory Manager" -W -h server.idm.example.com -p 389
Enter LDAP Password:
dn: cn=ipa_pwd_extop,cn=plugins,cn=config
```

```
changetype: modify
add: passSyncManagersDNs
passSyncManagersDNs: uid=admin,cn=users,cn=accounts,dc=example,dc=com
```

在 passSyncManagerDNs 下列出的 admin 用户现在具有额外的特权。

## 32.7. 检查 IDM 用户帐户是否已被锁住

作为身份管理(IdM)管理员，您可以检查 IdM 用户帐户是否已被锁住。为此，您必须将用户的最大允许失败的登录次数与用户实际失败的登录次数进行比较。

### 先决条件

- 您已在 IdM 中获得了管理用户的票据授予票(TGT)。

### 流程

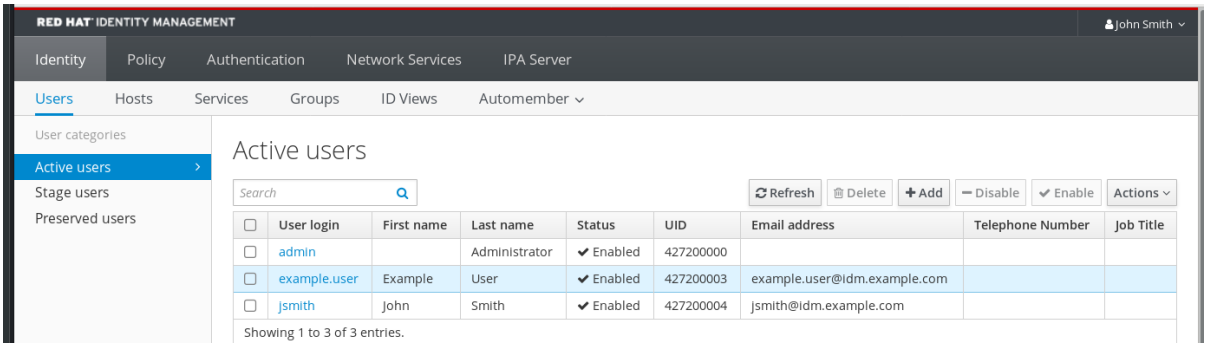
1. 显示用户帐户的状态，来查看失败的登录次数：

```
$ ipa user-status example_user
----------------------
Account disabled: False
----------------------
  Server: idm.example.com
  Failed logins: 8
  Last successful authentication: N/A
  Last failed authentication: 20220229080317Z
  Time now: 2022-02-29T08:04:46Z
---------------------------
Number of entries returned 1
---------------------------
```

2. 显示特定用户允许的登录尝试次数：

   a. 以 IdM 管理员身份登录到 IdM Web UI。

   b. 打开 Identity → Users → Active users 选项卡。




a. 点击用户名以打开用户设置。

b. 在 Password policy 部分中，找到 Max failures 项。

3. 将 ipa user-status 命令的输出中显示的失败的登录数与 IdM Web UI 中显示的 Max failures 数进行比较。如果失败的登录次数等于最大允许登录尝试次数，则用户帐户被锁住。

**其它资源**

- 在 IdM 中密码失败后解锁用户帐户

## 32.8. 在 IDM 中密码失败后解锁用户帐户

如果用户尝试使用不正确的密码进行一定次数的登录，则身份管理(IdM)会锁住用户帐户，从而阻止用户登录。出于安全考虑，IdM 不会显示用户帐户已被锁住的任何警告信息。相反，CLI 提示可能会一直要求用户输入密码。

IdM 在过了指定的时间后会自动解锁用户帐户。另外，您可以按照以下流程手动解锁用户帐户。

**先决条件**

- 您已获得 IdM 管理用户的票据授予票。

**流程**

- 要解锁用户帐户，请使用 ipa user-unlock 命令。

```
$ ipa user-unlock idm_user
----------------------
Unlocked account "idm_user"
----------------------
```

之后，用户可以再次登录。

**其它资源**

- 检查 IdM 用户帐户是否已被锁住

## 32.9. 为 IDM 中的用户启用最后一次成功 KERBEROS 验证的跟踪

出于性能方面的考虑，在 Red Hat Enterprise Linux 8 中运行的身份管理(IdM)不会存储用户最后一次成功的 Kerberos 验证的时间戳。因此，某些命令（如 ipa user-status ）不会显示时间戳。

**先决条件**

- 您已在 IdM 中获得了管理用户的票据授予票(TGT)。
- 您在执行该流程的 IdM 服务器上有 root 访问权限。

**流程**

1. 显示当前启用的密码插件功能：

```
# ipa config-show | grep "Password plugin features"
  Password plugin features: AllowNThash, KDC:Disable Last Success
```

输出显示 KDC:Disable Last Success 插件已启用。插件隐藏了最后一次成功的 Kerberos 身份验证，以防在 ipa user-status 输出中可见。

2. 将每个功能的 --ipaconfigstring=*feature* 参数添加到当前启用的 ipa config-mod 命令中，KDC:Disable Last Success 除外：

```
# ipa config-mod --ipaconfigstring='AllowNThash'
```

这个命令只启用 **AllowNThash** 插件。要启用多个功能，请为每个功能单独指定 **--ipaconfigstring=*feature*** 参数。

3. 重启 **IdM**:

```
# ipactl restart
```

# 第 33 章 定义 IDM 密码策略

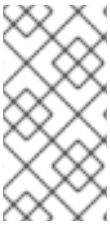
本章论述了 Identity Management (IdM) 密码策略，以及如何使用 Ansible playbook 在 IdM 中添加新的密码策略。

## 33.1. 什么是密码策略

密码策略是密码必须满足的一组规则。例如，password 策略可以定义最小密码长度和最大密码生命周期。受此策略影响的所有用户都必须设置足够长的密码，并经常更改密码以满足指定条件。这样，密码策略有助于降低某人发现和滥用用户密码的风险。

## 33.2. IDM 中的密码策略

密码是 Identity Management (IdM) 用户对 IdM Kerberos 域进行身份验证的最常用方式。密码策略定义了这些 IdM 用户密码必须满足的要求。



**注意**

IdM 密码策略在底层 LDAP 目录中设置，但 Kerberos 密钥分发中心 (KDC) 强制执行密码策略。

密码策略属性列出了您可以在 IdM 中定义密码策略的属性。

**表 33.1. 密码策略属性**

| 属性 | 介绍 | 示例 |
| --- | --- | --- |
| Max lifetime | 密码在必须重置密码之前有效的最长时间（以天为单位）。默认值为 90 天。<br><br>请注意，如果该属性被设为 0，则密码永远不会过期。 | Max lifetime = 180<br><br>用户密码仅有效 180 天。之后，IdM 会提示用户更改它们。 |
| Min lifetime | 两个密码更改操作之间必须经过的最短时间（以小时为单位）。 | Min Life = 1<br><br>用户更改密码后，他们必须至少等待 1 小时后再重新更改密码。 |
| History size | 保存的之前密码的数量。用户无法重复使用其密码历史记录中的密码，但可以重复利用未存储的旧密码。 | History size = 0<br><br>在这种情况下，密码历史记录为空，用户可以重复使用他们之前的任何密码。 |

| 属性 | 介绍 | 示例 |
| --- | --- | --- |
| Character classes | 用户必须在密码中使用的不同字符类别的数量。字符类为：<br><br>* 大写字符<br><br>* 小写字符<br><br>* 数字<br><br>* 特殊字符，如逗号(,)、句点(.)、星号(*)<br><br>* 其他 UTF-8 字符<br><br>当一个字符连续使用三次或更多次时，会将该字符类减一。例如：<br><br>* **Secret1** 有 3 个字符类：大写、小写、数字<br><br>* **Secret111** 具有 2 个字符类：大写、小写、数字以及重复使用 **1** 的 a -1 惩罚 | 字符类 = 0<br><br>需要的默认类数为 0。要配置数字，请使用 **--minclasses** 选项运行 **ipa pwpolicy-mod** 命令。<br><br>另请参阅此表下的 重要 备注。 |
| Min length | 密码中的最少字符数.<br><br>如果设置了任何 其他密码策略选项，则密码的最小长度为 6 个字符。 | Min length = 8<br><br>用户不能使用少于 8 个字符的密码。 |
| Max failures | IdM 锁定用户帐户前允许的失败登录的最多次数。 | Max failures = 6<br><br>当用户连续 7 次输入了错误的密码时，IdM 会锁定用户帐户。 |
| Failure reset interval | 在这个间隔后 IdM 重置当前失败登录尝试次数（以秒为单位）。 | Failure reset interval = 60<br><br>如果用户在 **Max failures** 定义的登录尝试失败的次数超过 1 分钟，用户可以尝试再次登录，而不会造成用户帐户锁定的风险。 |
| 锁定持续时间 | 在 **Max failures** 中定义的登录尝试失败次数后，用户帐户锁定的时间（以秒为单位）。 | Lockout duration = 600<br><br>锁定帐户的用户在 10 分钟内无法登录。 |

**重要**

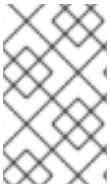
**如果您一组不同的硬件可能不能使用国际字符和符号，则字符类要求应为英语字母和常用符号。有关密码中字符类策略的更多信息，请参阅红帽知识库中的密码中哪些字符有效？**

## 33.3. 使用 ANSIBLE PLAYBOOK 在 IDM 中存在密码策略

按照以下流程，使用 Ansible playbook 确保密码策略在身份管理(IdM)中存在。

在 IdM 中的默认 global_policy 密码策略中，密码中不同字符类的数量设置为 0。历史记录大小也设置为 0。

完成此步骤，以使用 Ansible playbook 为 IdM 组强制执行更强大的密码策略。



**注意**

您只能为 IdM 组定义密码策略。您无法为单个用户定义密码策略。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。
- 正在确保 IdM 中存在密码策略的组。

**流程**

1. 创建一个清单文件，如 inventory.file，并在 [ipaserver] 部分中定义 IdM 服务器的 FQDN ：

```
[ipaserver]
server.idm.example.com
```

2. 创建 Ansible playbook 文件，以定义您要确保的密码策略。要简化此步骤，请复制并修改 /usr/share/doc/ansible-freeipa/playbooks/pwpolicy/pwpolicy_present.yml 文件中的示例：

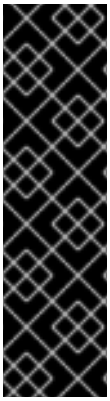
```
---
- name: Tests
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure presence of pwpolicy for group ops
    ipapwpolicy:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: ops
      minlife: 7
      maxlife: 49
      history: 5
      priority: 1
      lockouttime: 300
      minlength: 8
      minclasses: 4
      maxfail: 3
      failinterval: 5
```

有关单个变量含义的详情，请参阅密码策略属性。

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file
path_to_playbooks_directory_/new_pwpolicy_present.yml
```

您已成功使用 Ansible playbook 确保 IdM 中存在 ops 组的密码策略。



重要

ops 密码策略的优先级设置为 *1*，而 global_policy 密码策略没有设置优先级。因此，ops 策略会自动取代 ops 组的 global_policy，并立即强制执行。

当没有为用户设置任何组策略时，global_policy 充当备份策略，并且永远不会优先于组策略。

其它资源

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-pwpolicy.md 文件。
- 请参阅 密码策略优先级。

### 33.4. IDM 中的附加密码策略选项

作为身份管理 (IdM) 管理员，您可以通过启用基于 libpwquality 功能集的额外密码策略选项来增强默认密码要求。额外的密码策略选项包括：

**--maxrepeat**

指定新密码中相同连续字符的最大可接受数。

**--maxsequence**

指定新密码中单例字符序列的最大长度。此类序列的示例为 12345 或 fedcb。此类密码多数都不会通过简单检查。

**--dictcheck**

如果非零，则检查密码是否与字典中的词语匹配（如果可能修改）。目前，libpwquality 使用 cracklib 库执行字典检查。

**--usercheck**

如果非零，请检查密码是否以某种形式包含用户名，并可能进行修改。它不适用于少于 3 个字符的用户名。

您不能将额外的密码策略选项应用到现有密码。如果您应用了任何附加选项，IdM 会自动将 --minlength 选项（密码中的最少字符数）设置为 6 个字符。

**注意**

在使用 RHEL 7 和 RHEL 8 服务器的混合环境中，您只能在在 RHEL 8.4 及更新版本上运行的服务器中强制实施额外的密码策略设置。如果用户登录到 IdM 客户端，并且 IdM 客户端与运行在 RHEL 8.3 或更早版本上的 IdM 服务器进行通信，则系统管理员设置的新密码策略需求不会被应用。为确保行为的一致，请将所有服务器升级或更新至 RHEL 8.4 或更新的版本。

**其他资源：**

- 将额外密码策略应用到 IdM 组
- pwquality(3) man page

### 33.5. 将其他密码策略选项应用到 IDM 组

按照以下流程在身份管理(IdM)中应用额外的密码策略选项。这个示例描述了如何通过确保新密码不包含用户相应的用户名以及密码不包含两个以上相同的字符来增强 managers 组的密码策略。

**先决条件**

- 您以 IdM 管理员身份登录。
- managers 组存在于 IdM 中。
- IdM 中存在 managers 密码策略。

**流程**

1. 将用户名检查应用到 managers 组中用户建议的所有新密码：

```
$ ipa pwpolicy-mod --usercheck=True managers
```

**注意**

如果没有指定密码策略的名称，则会修改默认的 global_policy。

2. 在 manager 密码策略中，将相同连续字符的最大数量设置为 2：

```
$ ipa pwpolicy-mod --maxrepeat=2 managers
```

现在不接受包含连续两个以上相同字符的密码。例如，eR873mUi111YJQ 组合是不可接受的，因为它包含三个连续的 1。

验证

1. 添加名为 test_user 的测试用户：

```
$ ipa user-add test_user
First name: test
Last name: user
----------------------------
Added user "test_user"
----------------------------
```

2. 将 test 用户添加到 managers 组：

   a. 在 IdM Web UI 中，点 Identity → Groups → User Groups。

   b. 点 managers。

   c. 点 Add。

   d. 在 Add users to user group 'managers' 页面中，检查 test_user。

   e. 点击 > 箭头将用户移到 Prospect ive 列中。

   f. 点 Add。

3. 重置测试用户的密码：

a. 进入 **Identity → Users**。

b. 单击 **test_user**。

c. 在 **Actions** 菜单中，单击 **Reset Password**。

d. 输入用户的临时密码。

4. 在命令行中，尝试为 **test_user** 获取 **Kerberos** 票据授予票据 **(TGT)**：

```
$ kinit test_user
```

a. 输入临时密码。

b. 系统会通知您必须更改密码。输入包含用户名 **test_user** 的密码：

```
Password expired. You must change it now.
Enter new password:
Enter it again:
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.
```

**注意**

Kerberos 没有精细的错误密码策略报告，在某些情况下，没有提供拒绝密码的明确原因。

c. 系统通知您输入的密码被拒绝。输入包含连续三个或多个相同字符的密码：

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

d. 系统通知您输入的密码被拒绝。输入满足 managers 密码策略条件的密码：

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

5. 查看获取的 TGT:

```
$ klist
Ticket cache: KCM:0:33945
Default principal: test_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
07/07/2021 12:44:44  07/08/2021 12:44:44
krbtgt@IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
```

managers 密码策略现在可以为 managers 组中的用户正常工作。

其它资源

- IdM 中的额外密码策略

## 33.6. 使用 ANSIBLE PLAYBOOK 将额外的密码策略选项应用到 IDM 组

您可以使用 Ansible playbook 应用额外的密码策略选项，以加强特定的 IdM 组所要求的密码策略。为此，您可以使用 maxrepeat、maxsequence、dictcheck 和 usercheck 密码策略选项。这个示例描述了如何为 managers 组设置以下要求：

- 用户的新密码不包含用户各自的用户名。
- 密码不包含连续两个以上相同的字符。
- 密码中的任何单调字符序列不超过 3 个字符。这意味着系统不接受具有类似序列（如 1234 或 abcd） 的密码。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。
- 正在确保 IdM 中存在密码策略的组。

**流程**

1. 创建 Ansible playbook 文件 manager_pwpolicy_present.yml，其定义您要确保其存在的密码策略。要简化此步骤，请复制并修改以下示例：

```
---
- name: Tests
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure presence of usercheck and maxrepeat pwpolicy for group managers
    ipapwpolicy:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: managers
      usercheck: True
      maxrepeat: 2
      maxsequence: 3
```

2. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file
path_to_playbooks_directory_/manager_pwpolicy_present.yml
```

验证

1. 添加名为 test_user 的测试用户：

```
$ ipa user-add test_user
First name: test
Last name: user
---------------------------
Added user "test_user"
---------------------------
```

2. 将 test 用户添加到 managers 组：

   a. 在 IdM Web UI 中，点 Identity → Groups → User Groups。

   b. 点 managers。

   c. 点 Add。

   d. 在 Add users to user group 'managers' 页面中，检查 test_user。

   e. 点击 > 箭头将用户移到 Prospect ive 列中。

   f. 点 Add。

3. 重置测试用户的密码：

   a. 进入 Identity → Users。

   b. 单击 test_user。

   c. 在 Actions 菜单中，单击 Reset Password。

d. 输入用户的临时密码。

4. 在命令行中，尝试为 test_user 获取 Kerberos 票据授予票据 (TGT)：

```
$ kinit test_user
```

a. 输入临时密码。

b. 系统会通知您必须更改密码。输入包含用户名 test_user 的密码：

```
Password expired. You must change it now.
Enter new password:
Enter it again:
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.
```

**注意**

Kerberos 没有精细的错误密码策略报告，在某些情况下，没有提供拒绝密码的明确原因。

c. 系统通知您输入的密码被拒绝。输入包含连续三个或多个相同字符的密码：

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

d. 系统通知您输入的密码被拒绝。输入一个包含超过 3 个字符长的单调字符序列的密码。此类序列的示例包括 1234 和 fedc ：

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

e. 系统通知您输入的密码被拒绝。输入满足 managers 密码策略条件的密码：

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

5. 验证您是否已获得 TGT，这只有在输入有效密码后才有可能：

```
$ klist
Ticket cache: KCM:0:33945
Default principal: test_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
07/07/2021 12:44:44  07/08/2021 12:44:44
krbtgt@IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
```

**其它资源**

- IdM 中的额外密码策略
- /usr/share/doc/ansible-freeipa/README-pwpolicy.md
- /usr/share/doc/ansible-freeipa/playbooks/pwpolicy

# 第 34 章 管理过期密码通知

您可以使用 ipa-client-epn 软件包提供的过期密码通知(EPN)工具来构建一个身份管理(IdM)用户列表，这些用户的密码在配置的时间内即将过期。要安装、配置和使用 EPN 工具，请参阅相关章节。

- 什么是过期的密码通知工具
- 安装过期的密码通知工具
- 运行 EPN 工具，向密码即将过期的用户发送电子邮件
- 启用 ipa-epn.timer ，向密码即将过期的所有用户发送电子邮件
- 修改过期密码通知电子邮件模板

## 34.1. 什么是过期的密码通知工具

过期密码通知(EPN)工具是一个独立的工具，可用于构建一个身份管理(IdM)用户列表，这些用户的密码在配置的时间内即将过期。

IdM 管理员可以使用 EPN 进行以下操作：

- 以 JSON 格式显示受影响的用户的列表，该列表是在dry-run 模式下运行时创建的。
- 计算在给定日期或日期范围内发送多少封电子邮件.
- 向用户发送密码过期电子邮件通知.
- 将 ipa-epn.timer 配置为每天运行 EPN 工具，并向密码在定义的未来日期范围内即将过期的用户发送电子邮件。

- 自定义要发送给用户的电子邮件通知。

**注意**

如果用户帐户被禁用，则不会发送电子邮件通知（如果密码即将过期）。

## 34.2. 安装过期的密码通知工具

按照以下流程安装过期密码通知(EPN)工具。

**先决条件**

- 在身份管理(IdM)副本或配置为智能主机的本地 Postfix SMTP 服务器的 IdM 客户端上安装 EPN 工具。

**流程**

- 安装 EPN 工具：

```
# yum install ipa-client-epn
```

## 34.3. 运行 EPN 工具，向密码即将过期的用户发送电子邮件

按照以下流程运行过期密码通知(EPN)工具，向密码即将过期的用户发送电子邮件。

**注意**

EPN 工具是无状态的。如果 EPN 工具未能向密码即将在给定日期过期的任何用户发送邮件，则 EPN 工具不会保存这些用户的列表。

**先决条件**

- ipa-client-epn 软件包已安装。请参阅 安装过期密码通知工具。
- 如果需要，自定义 ipa-epn 电子邮件模板。请参阅 修改过期密码通知电子邮件模板。

**流程**

1. 更新 epn.conf 配置文件，来为 EPN 工具设置选项，以通知用户密码即将过期。

```
# vi /etc/ipa/epn.conf
```

2. 根据需要更新 notify_ttls。默认是通知用户其密码将在 28、14、7、3 和 1 天后过期。

```
notify_ttls = 28, 14, 7, 3, 1
```

3. 配置 SMTP 服务器和端口：

```
smtp_server = localhost
smtp_port = 25
```

4. 指定发送电子邮件过期通知的电子邮件地址。任何未成功发送的电子邮件都将返回到此地址。

```
mail_from =admin-email@example.com
```

5. 保存 /etc/ipa/epn.conf 文件。

6. 以 dry-run 模式运行 EPN 工具，来生成一个用户列表，如果您不使用 --dry-run 选项来运行工具，则密码过期电子邮件通知将发送给这些用户。

```
ipa-epn --dry-run
[
   {
    "uid": "user5",
    "cn": "user 5",
    "krbpasswordexpiration": "2020-04-17 15:51:53",
    "mail": "['user5@ipa.test']"
   }
]
[
   {
    "uid": "user6",
    "cn": "user 6",
    "krbpasswordexpiration": "2020-12-17 15:51:53",
    "mail": "['user5@ipa.test']"
    }
]
The IPA-EPN command was successful
```

**注意**

如果返回的用户列表非常大，并且运行工具时没有 **--dry-run** 选项，这可能会导致您的电子邮件服务器出现问题。

7. 不使用 **--dry-run** 选项运行 **EPN** 工具，来将到期电子邮件发送给当您在 **dry-run** 模式下运行 **EPN** 工具时返回的所有用户的列表：

```
ipa-epn
[
 {
   "uid": "user5",
   "cn": "user 5",
   "krbpasswordexpiration": "2020-10-01 15:51:53",
   "mail": "['user5@ipa.test']"
 }
]
[
 {
  "uid": "user6",
  "cn": "user 6",
  "krbpasswordexpiration": "2020-12-17 15:51:53",
  "mail": "['user5@ipa.test']"
 }
]
The IPA-EPN command was successful
```

8. 您可以将 **EPN** 添加到任何监控系统，并使用 **--from-nbdays** 和 **--to-nbdays** 选项调用它，以确定在特定时间范围内将有多少个用户的密码即将过期：

```
# ipa-epn --from-nbdays 8 --to-nbdays 12
```

**注意**

如果您使用 **--from-nbdays** 和 **--to-nbdays** 选项调用 **EPN** 工具，它将自动在 **dry-run** 模式下执行。

**验证步骤**

- 运行 **EPN** 工具，并验证是否已发送电子邮件通知。

**其它资源**

- 请参阅 ipa-epn 手册页。

- 请参阅 epn.conf 手册页。

### 34.4. 启用 IPA-EPN.TIMER ，向密码即将过期的所有用户发送电子邮件

按照以下流程，使用 ipa-epn.timer 运行过期密码通知(EPN)工具，向密码即将过期的用户发送电子邮件。ipa-epn.timer 解析 epn.conf 文件，并向在该文件中配置的定义的未来日期范围内密码即将过期的用户发送电子邮件。

**先决条件**

- ipa-client-epn 软件包已安装。请参阅 安装过期密码通知工具

- 如果需要，自定义 ipa-epn 电子邮件模板。请参阅 修改过期密码通知电子邮件模板

**流程**

- 启动 ipa-epn.timer:

```
systemctl start ipa-epn.timer
```

启动计时器后，默认情况下 EPN 工具会在每天早晨 1 点运行。

**其它资源**

- 请参阅 ipa-epn 手册页。

### 34.5. 修改过期密码通知电子邮件模板

按照以下流程自定义过期密码通知(EPN)电子邮件消息模板。

**先决条件**

- ipa-client-epn 软件包已安装。

流程

1. 打开 EPN 消息模板：

```
# vi /etc/ipa/epn/expire_msg.template
```

2. 根据需要更新模板文本。

```
Hi {{ fullname }},

Your password will expire on {{ expiration }}.

Please change it as soon as possible.
```

您可以在模板中使用以下变量：

- 用户 ID：uid
- 全名：fullname
- 名字：first
- 姓氏：last
- 密码过期日期：过期

3. 保存消息模板文件。

验证步骤

- 运行 EPN 工具，并验证电子邮件通知包含更新的文本。

**其它资源**

- 请参阅 **ipa-epn** 手册页。

# 第 35 章 使用 ID 视图来覆盖 IDM 客户端上的用户属性值

如果身份管理(IdM)用户想要覆盖存储在 IdM LDAP 服务器中的某些用户或组属性，如登录名称、主目录、用于身份验证的证书或 SSH 密钥，则您作为 IdM 管理员可使用 IdM ID 视图重新定义特定 IdM 客户端上的这些值。例如，您可以为用户最常用于登录 IdM 的 IdM 客户端上为用户指定不同的主目录。

本章描述了如何重新定义与作为客户端注册到 IdM 的主机上的 IdM 用户关联的 POSIX 属性值。

## 35.1. ID 视图

身份管理(IdM)中的 ID 视图是一个指定以下信息的 IdM 客户端视图：

- 集中定义的 POSIX 用户或组属性的新值
- 应用新值的客户端主机或主机。

ID 视图包含一个或多个覆盖。覆盖是集中定义的 POSIX 属性值的特定替换。

您只能为集中在 IdM 服务器上的 IdM 客户端定义 ID 视图。您无法为本地 IdM 客户端配置客户端覆盖。

例如，您可以使用 ID 视图来实现以下目标：

- 为不同的环境定义不同的属性值。例如，您可以允许 IdM 管理员或其他 IdM 用户在不同的 IdM 客户端上拥有不同的主目录：您可以将 /home/crypt/username 配置为此用户在一个 IdM 客户端上的主目录，将 /dropbox/username 配置为此用户在另一个客户端上的主目录。在这种情况下使用 ID 视图非常方便，例如，更改客户端 /etc/sssd/sssd.conf 文件中的 fallback_homedir、overwrite_homedir 或其他主目录变量将影响所有用户。有关示例过程，请参阅 添加 ID 视图来覆盖 IdM 客户端上的 IdM 用户主目录。
- 将之前生成的属性值替换为其他值，例如覆盖用户的 UID。当您要实现系统范围的更改时，此功能非常有用，否则在 LDAP 端很难实现，例如将 1009 设为 IdM 用户的 UID。用于生成 IdM 用户 UID 的 IdM ID 范围一开始不要低于 1000 甚至 10000。如果 IdM 用户在所有 IdM 客户端上模拟 UID 为1009 的本地用户是有原因的，那么您可以使用 ID 视图覆盖在 IdM 中创建用户时生成的 IdM 用户的 UID。

重要

您只能将 ID 视图应用于 IdM 客户端，不能应用于 IdM 服务器。

其它资源

- 为活动目录用户使用 ID 视图
- SSSD 客户端视图

## 35.2. ID 视图对 SSSD 性能的潜在负面影响

当您定义 ID 视图时，IdM 会将所需的覆盖值放在 IdM 服务器的系统安全服务守护进程(SSSD)缓存中。在 IdM 客户端上运行的 SSSD 然后从服务器缓存中检索覆盖值。

应用 ID 视图可能会对系统安全服务守护进程(SSSD)的性能造成负面影响，因为某些优化和 ID 视图无法同时运行。例如，ID 视图会防止 SSSD 优化在服务器上查找组的过程：

- 使用 ID 视图时，如果组名称已被覆盖，SSSD 必须检查返回的组成员名称列表中的每个成员。
- 如果没有 ID 视图，SSSD 只能从组对象的成员属性收集用户名。

当 SSSD 缓存为空或清除缓存后，这种负面影响变得非常明显，使得所有条目都无效。

## 35.3. ID 视图可以覆盖的属性

ID 视图由用户和组 ID 覆盖组成。覆盖定义新的 POSIX 属性值。

用户和组 ID 覆盖可以为以下 POSIX 属性定义新值：

用户属性

- 登录名(uid)
- GECOS 条目(gecos)
- UID 号(uidNumber)
- GID 号(gidNumber)
- 登录 shell(loginShell)
- 主目录（homeDirectory）
- SSH 公钥(ipaSshPubkey)
- 证书(userCertificate)

组属性

- 组名(cn)
- 组 GID 号(gidNumber)

### 35.4. 获取 ID 视图命令的帮助信息

您可以获得 IdM 命令行界面(CLI)上涉及身份管理(IdM)ID 视图的命令的帮助。

先决条件

- 您已获得了 IdM 用户的 Kerberos 票据。

流程

- 要显示用于管理 ID 视图和覆盖的所有命令：

```
$ ipa help idviews
ID Views

Manage ID Views

IPA allows to override certain properties of users and groups[...]
[...]
Topic commands:
  idoverridegroup-add          Add a new Group ID override
  idoverridegroup-del          Delete a Group ID override
[...]
```

- 要显示特定命令的详细帮助信息，请在命令中添加 --help 选项：

```
$ ipa idview-add --help
Usage: ipa [global-options] idview-add NAME [options]

Add a new ID View.
Options:
  -h, --help      show this help message and exit
  --desc=STR      Description
[...]
```

## 35.5. 使用 ID 视图来覆盖特定主机上 IDM 用户的登录名称

按照以下流程，为特定 IdM 客户端创建一个 ID 视图，该视图覆盖与特定 IdM 用户关联的 POSIX 属性值。该流程使用 ID 视图示例，它可让名为 idm_user 的 IdM 用户使用 user_1234 登录名称登录到名为 host1 的 IdM 客户端。

先决条件

- 以 IdM 管理员身份登录。

流程

1. 创建新的 ID 视图。例如，创建名为 example_for_host1 的 ID 视图：

```
$ ipa idview-add example_for_host1
---------------------------
Added ID View "example_for_host1"
```

```
---------------------------
 ID View Name: example_for_host1
```

2. 将用户覆盖添加到 example_for_host1 ID 视图。覆盖用户登录：

   - 输入 ipa idoverrideuser-add 命令
   - 添加 ID 视图的名称
   - 添加用户名，也称为锚
   - 添加 --login 选项：

```
$ ipa idoverrideuser-add example_for_host1 idm_user --login=user_1234
----------------------------
Added User ID override "idm_user"
----------------------------
  Anchor to override: idm_user
  User login: user_1234
```

   要获得可用选项列表，请运行 ipa idoverrideuser-add --help。

   **注意**

   ipa idoverrideuser-add --certificate 命令替换指定 ID 视图中帐户的所有现有证书。要附加额外的证书，请使用 ipa idoverrideuser-add-cert 命令：

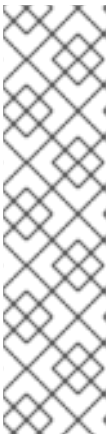
```
$ ipa idoverrideuser-add-cert example_for_host1 user --
certificate="MIIEATCC..."
```

3. 可选：使用 ipa idoverrideuser-mod 命令，您可以为现有用户覆盖指定新的属性值。

4. 将 example_for_host1 应用到 host1.idm.example.com 主机：

```
$ ipa idview-apply example_for_host1 --hosts=host1.idm.example.com
```

```
----------------------------
Applied ID View "example_for_host1"
----------------------------
hosts: host1.idm.example.com
------------------------------------------
Number of hosts the ID View was applied to: 1
------------------------------------------
```



**注意**

**ipa idview-apply 命令也接受 --hostgroups 选项。选项将 ID 视图应用到属于指定主机组的主机，但不会将 ID 视图与主机组本身相关联。相反，--hostgroups 选项会展开指定主机组的成员，并将 --hosts 选项分别应用到其中的每一个成员。**

**这意味着，如果以后将主机添加到主机组中，则 ID 视图不会应用到新主机。**

5. **要将新配置立即应用到 host1.idm.example.com 系统：**

   a. **以 root 身份通过 SSH 连接到系统：**

   ```
   $ ssh root@host1
   Password:
   ```

   b. **清除 SSSD 缓存：**

   ```
   root@host1 ~]# sss_cache -E
   ```

   c. **重启 SSSD 守护进程：**

```
root@host1 ~]# systemctl restart sssd
```

**验证步骤**

- **如果您有 user_1234 的凭证，您可以使用它们登录到 host1 上的 IdM：**

  1. **使用 user_1234 作为登录名称，通过 SSH 连接到 host1 ：**

```
[root@r8server ~]# ssh user_1234@host1.idm.example.com
Password:

Last login: Sun Jun 21 22:34:25 2020 from 192.168.122.229
[user_1234@host1 ~]$
```

2. 显示工作目录：

```
[user_1234@host1 ~]$ pwd
/home/idm_user/
```

- 或者，如果您在 host1 上有 root 凭证，您可以使用它们来检查 idm_user 和 user_1234 的 id 命令的输出：

```
[root@host1 ~]# id idm_user
uid=779800003(user_1234) gid=779800003(idm_user) groups=779800003(idm_user)
[root@host1 ~]# user_1234
uid=779800003(user_1234) gid=779800003(idm_user) groups=779800003(idm_user)
```

### 35.6. 修改 IDM ID 视图

身份管理(IdM)中的 ID 视图覆盖与特定 IdM 用户关联的 POSIX 属性值。按照以下流程修改现有 ID 视图。具体来说，它描述了如何修改 ID 视图以使名为 idm_user 的用户使用 /home/user_1234/ 目录作为用户主目录，而不是使用 host1.idm.example.com IdM 客户端上的 /home/idm_user/。

**先决条件**

- 具有对 host1.idm.example.com 的 root 访问权限。
- 您已以具有所需特权的用户身份登录，如 admin。
- 您为 idm_user 配置了一个 ID 视图，它适用于 host1 IdM 客户端。

**流程**

1. 以 root 用户身份，创建您希望 idm_user 在 host1.idm.example.com 上作为用户主目录使用的目录：

```
[root@host1 /]# mkdir /home/user_1234/
```

2. 更改目录的所有权：

```
[root@host1 /]# chown idm_user:idm_user /home/user_1234/
```

3. 显示 ID 视图，包括当前要应用 ID 视图的主机。显示名为 example_for_host1 的 ID 视图：

```
$ ipa idview-show example_for_host1 --all
  dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
  ID View Name: example_for_host1
  User object override: idm_user
  Hosts the view applies to: host1.idm.example.com
  objectclass: ipaIDView, top, nsContainer
```

输出显示 ID 视图当前应用于 host1.idm.example.com。

4. 修改 example_for_host1 ID 视图的用户覆盖。覆盖用户主目录：

- 输入 ipa idoverrideuser-add 命令
- 添加 ID 视图的名称
- 添加用户名，也称为锚
- 添加 --homedir 选项：

```
$ ipa idoverrideuser-mod example_for_host1 idm_user --
homedir=/home/user_1234
----------------------------
Modified a User ID override "idm_user"
----------------------------
  Anchor to override: idm_user
  User login: user_1234
  Home directory: /home/user_1234/
```

要获得可用选项的列表，请运行 ipa idoverrideuser-mod --help。

5. 要将新配置立即应用到 host1.idm.example.com 系统：

   a. 以 root 身份通过 SSH 连接到系统：

   ```
   $ ssh root@host1
   Password:
   ```

   b. 清除 SSSD 缓存：

   ```
   root@host1 ~]# sss_cache -E
   ```

   c. 重启 SSSD 守护进程：

   ```
   root@host1 ~]# systemctl restart sssd
   ```

**验证步骤**

1. 以 idm_user用户身份，通过 SSH 连接到 host1:

   ```
   [root@r8server ~]# ssh idm_user@host1.idm.example.com
   Password:

   Last login: Sun Jun 21 22:34:25 2020 from 192.168.122.229
   [user_1234@host1 ~]$
   ```

2. 打印工作目录：

   ```
   [user_1234@host1 ~]$ pwd
   /home/user_1234/
   ```

**其它资源**

- 通过修改 Default Trust View 为 AD 用户定义全局属性

## 35.7. 添加 ID 视图来覆盖 IDM 客户端上的 IDM 用户主目录

身份管理(IdM)中的 ID 视图覆盖与特定 IdM 用户关联的 POSIX 属性值。按照以下流程，在名为 host1 的 IdM 客户端上创建一个应用到 idm_user 的 ID 视图，以允许用户将 /home/user_1234/ 目录用作用户

主目录，而不是 /home/idm_user/。

**先决条件**

- 具有对 host1.idm.example.com 的 root 访问权限。
- 您已以具有所需特权的用户身份登录，如 admin。

**流程**

1. 以 root 用户身份，创建您希望 idm_user 在 host1.idm.example.com 上作为用户主目录使用的目录：

```
[root@host1 /]# mkdir /home/user_1234/
```

2. 更改目录的所有权：

```
[root@host1 /]# chown idm_user:idm_user /home/user_1234/
```

3. 创建 ID 视图。例如，创建名为 example_for_host1 的 ID 视图：

```
$ ipa idview-add example_for_host1
---------------------------
Added ID View "example_for_host1"
---------------------------
  ID View Name: example_for_host1
```

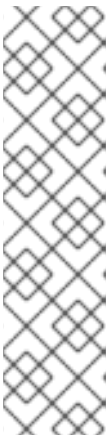
4. 将用户覆盖添加到 example_for_host1 ID 视图。覆盖用户主目录：
   - 输入 ipa idoverrideuser-add 命令
   - 添加 ID 视图的名称
   - 添加用户名，也称为锚

- 添加 --homedir 选项：

```
$ ipa idoverrideuser-add example_for_host1 idm_user --homedir=/home/user_1234
---------------------------
Added User ID override "idm_user"
---------------------------
  Anchor to override: idm_user
  Home directory: /home/user_1234/
```

5. 将 example_for_host1 应用到 host1.idm.example.com 主机：

```
$ ipa idview-apply example_for_host1 --hosts=host1.idm.example.com
---------------------------
Applied ID View "example_for_host1"
---------------------------
hosts: host1.idm.example.com
-----------------------------------------
Number of hosts the ID View was applied to: 1
-----------------------------------------
```



**注意**

ipa idview-apply 命令也接受 --hostgroups 选项。选项将 ID 视图应用到属于指定主机组的主机，但不会将 ID 视图与主机组本身相关联。相反，--hostgroups 选项会展开指定主机组的成员，并将 --hosts 选项分别应用到其中的每一个成员。

这意味着，如果以后将主机添加到主机组中，则 ID 视图不会应用到新主机。

6. 要将新配置立即应用到 host1.idm.example.com 系统：

   a. 以 root 身份通过 SSH 连接到系统：

```
$ ssh root@host1
Password:
```

   b. 清除 SSSD 缓存：

```
root@host1 ~]# sss_cache -E
```

c. 重启 SSSD 守护进程：

```
root@host1 ~]# systemctl restart sssd
```

验证步骤

1. 以 idm_user用户身份，通过 SSH 连接到 host1:

```
[root@r8server ~]# ssh idm_user@host1.idm.example.com
Password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sun Jun 21 22:34:25 2020 from 192.168.122.229
[idm_user@host1 /]$
```

2. 打印工作目录：

```
[idm_user@host1 /]$ pwd
/home/user_1234/
```

其它资源

- 对带有 ID 视图的 IdM 客户端上的 AD 用户覆盖 Default Trust View 属性

## 35.8. 将 ID 视图应用到 IDM 主机组

ipa idview-apply 命令接受 --hostgroups 选项。不过，选项充当一次性操作，它将 ID 视图应用到当前属于指定主机组的主机，但不动态地将 ID 视图与主机组本身关联。--hostgroups 选项将展开指定主机组的成员，并将 --hosts 选项分别应用到其中的每一个成员。

如果稍后向主机组添加新主机，您必须使用 ipa idview-apply 命令及 --hosts 选项，手动将 ID 视图应用到新主机。

类似地，如果您从主机组中删除主机，则移除后 ID 视图仍会分配给该主机。要从删除的主机中取消 ID 视图应用，您必须运行 ipa idview-unapply *id_view_name* --hosts=*name_of_the_removed_host* 命令。

按照以下流程实现以下目标：

1. 如何创建主机组并向其添加主机。

2. 如何将 ID 视图应用到主机组。

3. 如何向主机组添加新主机，并将 ID 视图应用到新主机。

**先决条件**

- 确保 IdM 中存在您要应用到主机组的 ID 视图。例如：要创建一个 ID 视图来覆盖特定 IdM 客户端上的 IdM 用户登录名称，请参阅 使用 ID 视图覆盖特定主机上 IdM 用户的登录名称。

**流程**

1. 创建主机组并为其添加主机：

   a. 创建主机组.例如，创建名为 baltimore 的主机组：

   ```
   [root@server ~]# ipa hostgroup-add --desc="Baltimore hosts" baltimore
   ---------------------------
   Added hostgroup "baltimore"
   ---------------------------
   Host-group: baltimore
   Description: Baltimore hosts
   ```

   b. 将主机添加到主机组。例如，将 host102 和 host103 添加到 baltimore 主机组：

   ```
   [root@server ~]# ipa hostgroup-add-member --hosts={host102,host103} baltimore
   Host-group: baltimore
   Description: Baltimore hosts
   Member hosts: host102.idm.example.com, host103.idm.example.com
   -------------------------
   Number of members added 2
   -------------------------
   ```

2. 将 ID 视图应用到主机组中的主机。例如，要将 example_for_host1 ID 视图应用到 baltimore 主机组：

   ```
   [root@server ~]# ipa idview-apply --hostgroups=baltimore
   ID View Name: example_for_host1
   ```

```
---------------------------------------
Applied ID View "example_for_host1"
---------------------------------------
  hosts: host102.idm.example.com, host103.idm.example.com
-----------------------------------------
Number of hosts the ID View was applied to: 2
-----------------------------------------
```

3. 将新主机添加到主机组，并将 ID 视图应用到新主机：

   a. 将新主机添加到主机组。例如，要将 somehost.idm.example.com 主机添加到 baltimore 主机组：

   ```
   [root@server ~]# ipa hostgroup-add-member --hosts=somehost.idm.example.com
   baltimore
     Host-group: baltimore
     Description: Baltimore hosts
     Member hosts:  host102.idm.example.com,
   host103.idm.example.com,somehost.idm.example.com
   -------------------------
   Number of members added 1
   -------------------------
   ```

   b. （可选）显示 ID 视图信息。例如，要显示 example_for_host1 ID 视图的详情：

   ```
   [root@server ~]# ipa idview-show example_for_host1 --all
     dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
     ID View Name: example_for_host1
   [...]
     Hosts the view applies to: host102.idm.example.com, host103.idm.example.com
     objectclass: ipaIDView, top, nsContainer
   ```

   输出显示 ID 视图没有应用到 somehost.idm.example.com（在 baltimore 主机组中新添加的主机）。

   c. 将 ID 视图应用到新主机。例如，要将 example_for_host1 ID 视图应用到 somehost.idm.example.com ：

   ```
   [root@server ~]# ipa idview-apply --host=somehost.idm.example.com
   ID View Name: example_for_host1
   ---------------------------------------
   Applied ID View "example_for_host1"
   ---------------------------------------
     hosts: somehost.idm.example.com
   ```

```
----------------------------------------
Number of hosts the ID View was applied to: 1
----------------------------------------
```

验证步骤

- 再次显示 ID 视图信息：

```
[root@server ~]# ipa idview-show example_for_host1 --all
  dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
  ID View Name: example_for_host1
[...]
  Hosts the view applies to: host102.idm.example.com, host103.idm.example.com,
somehost.idm.example.com
  objectclass: ipaIDView, top, nsContainer
```

输出显示 ID 视图现在已应用到 somehost.idm.example.com（在 baltimore 主机组中新添加的主机）。

## 35.9. 使用 ANSIBLE 覆盖特定主机上 IDM 用户的登录名称和主目录

完成此流程，以使用 idoverrideuser ansible-freeipa 模块为特定身份管理(IdM)客户端创建 ID 视图，以覆盖与特定 IdM 用户关联的 POSIX 属性值。该流程使用 ID 视图的示例，该视图可让名为 idm_user 的 IdM 用户使用 user_1234 登录名称登录到名为 host1.idm.example.com 的 IdM 客户端。此外，ID 视图会修改 idm_user 的主目录，以便在登录 host1 后，用户主目录为 /home/user_1234/。

先决条件

- 在控制节点上：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已安装 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您使用 RHEL 8.10 或更高版本。

- 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 使用以下内容创建 Ansible playbook 文件 add-idoverrideuser-with-name-and-homedir.yml ：

```
---
- name: Playbook to manage idoverrideuser
  hosts: ipaserver
  become: false
  gather_facts: false
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Ensure idview_for_host1 is present
    idview:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: idview_for_host1
  - name: Ensure idview_for_host1 is applied to host1.idm.example.com
    idview:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: idview_for_host1
      host: host1.idm.example.com
      action: member
  - name: Ensure idm_user is present in idview_for_host1 with homedir
/home/user_1234 and name user_1234
    ipaidoverrideuser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      idview: idview_for_host1
      anchor: idm_user
      name: user_1234
      homedir: /home/user_1234
```

2. 运行 playbook。指定 playbook 文件，存储保护 secret.yml 文件的密码，以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/inventory <path_to_playbooks_directory>/add-
idoverrideuser-with-name-and-homedir.yml
```

3. [可选] 如果您有 root 凭证，您可以立即将新配置应用到 host1.idm.example.com 系统：

a. 以 root 身份通过 SSH 连接到系统：

```
$ ssh root@host1
Password:
```

b. 清除 SSSD 缓存：

```
root@host1 ~]# sss_cache -E
```

c. 重启 SSSD 守护进程：

```
root@host1 ~]# systemctl restart sssd
```

验证

1. 以 idm_user用户身份，通过 SSH 连接到 host1:

```
[root@r8server ~]# ssh idm_user@host1.idm.example.com
Password:

Last login: Sun Jun 21 22:34:25 2020 from 192.168.122.229
[user_1234@host1 ~]$
```

2. 打印工作目录：

```
[user_1234@host1 ~]$ pwd
/home/user_1234/
```

其它资源

- ansible-freeipa 上游文档中的 idoverrideuser 模块

### 35.10. 使用 ANSIBLE 配置在 IDM 客户端上启用 SSH 密钥登录的 ID 视图

完成此流程，以使用 idoverrideuser ansible-freeipa 模块来确保 IdM 用户可以使用特定的 SSH 密钥登录到特定的 IdM 客户端。该流程使用 ID 视图的示例，它可让名为 idm_user 的 IdM 用户使用 SSH 密钥登录到名为 host1.idm.example.com 的 IdM 客户端。

注意

此 ID 视图可用于增强特定的 HBAC 规则。

先决条件

- 在控制节点上：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已安装 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您使用 RHEL 8.10 或更高版本。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。
- 您可以访问 idm_user的 SSH 公钥。
- idview_for_host1 ID 视图存在。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

流程

1. 使用以下内容创建 Ansible playbook 文件 ensure-idoverrideuser-can-login-with-sshkey.yml ：

```
---
- name: Playbook to manage idoverrideuser
  hosts: ipaserver
```

```
  become: false
  gather_facts: false
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Ensure test user idm_user is present in idview idview_for_host1 with
sshpubkey
    ipaidoverrideuser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      idview: idview_for_host1
      anchor: idm_user
      sshpubkey:
      - ssh-rsa AAAAB3NzaC1yc2EAAADAQABAAABgQCqmVDpEX5gnSjKuv97Ay ...
  - name: Ensure idview_for_host1 is applied to host1.idm.example.com
    ipaidview:
      ipaadmin_password:  "{{ ipaadmin_password }}"
      name: idview_for_host1
      host: host1.idm.example.com
      action: member
```

2. 运行 playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/inventory <path_to_playbooks_directory>/ensure-
idoverrideuser-can-login-with-sshkey.yml
```

3. [可选] 如果您有 root 凭证，您可以立即将新配置应用到 host1.idm.example.com 系统：

   a. 以 root 身份通过 SSH 连接到系统：

   ```
   $ ssh root@host1
   Password:
   ```

   b. 清除 SSSD 缓存：

   ```
   root@host1 ~]# sss_cache -E
   ```

   c. 重启 SSSD 守护进程：

   ```
   root@host1 ~]# systemctl restart sssd
   ```

验证

- 使用 SSH 到 host1 的公钥：

```
[root@r8server ~]# ssh -i ~/.ssh/id_rsa.pub idm_user@host1.idm.example.com

Last login: Sun Jun 21 22:34:25 2023 from 192.168.122.229
[idm_user@host1 ~]$
```

输出确认您已成功登录。

其它资源

- ansible-freeipa 上游文档中的 idoverrideuser 模块

## 35.11. 使用 ANSIBLE 为用户提供 ID 覆盖对 IDM 客户端上本地声音卡的访问权限

您可以使用 ansible-freeipa 组和 idoverrideuser 模块在 IdM 客户端上使身份管理(IdM)或 Active Directory (AD)用户成员。这会授予 IdM 或 AD 用户对主机上声音卡的特权访问权限。该流程使用 Default Trust View ID 视图的示例，在第一个 playbook 任务中添加 aduser@addomain.com ID 覆盖。在下一个 playbook 任务中，在 IdM 中创建 音频 组，GID 为 63，它对应于 RHEL 主机上的本地 音频 组的 GID。同时，aduser@addomain.com ID 覆盖作为成员添加到 IdM 音频组中。

先决条件

- 您有访问要在其上执行流程第一部分的 IdM 客户端的 root 访问权限。在示例中，这是 client.idm.example.com。

- 您已配置了 Ansible 控制节点以满足以下要求：

  - 您使用 Ansible 版本 2.14 或更高版本。

  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。

  - 您使用 RHEL 8.10 或更高版本。

  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- AD 林与 IdM 信任。在示例中，AD 域的名称是 addomain.com，以及 AD 用户的完全限定域名(FQDN)，其存在于本地 音频 组中存在是 aduser@addomain.com。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 在 client.idm.example.com 上，将 [SUCCESS=merge] 添加到 /etc/nsswitch.conf 文件中：

```
[...]
# Allow initgroups to default to the setting for group.
initgroups: sss [SUCCESS=merge] files
```

2. 确定本地 音频 组的 GID：

```
$ getent group audio
--------------------
audio:x:63
```

3. 在 Ansible 控制节点上，创建一个带有任务的 add-aduser-to-audio-group.yml playbook，将 aduser@addomain.com 用户覆盖添加到 Default Trust View 中：

```
---
- name: Playbook to manage idoverrideuser
  hosts: ipaserver
  become: false

  tasks:
  - name: Add aduser@addomain.com user to the Default Trust View
    ipaidoverrideuser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      idview: "Default Trust View"
      anchor: aduser@addomain.com
```

4. 在同一 playbook 中使用另一个 playbook 任务，将组 音频 添加到 IdM 中，GID 为 63。将 aduser idoverrideuser 添加到组中：

```
 - name: Add the audio group with the aduser member and GID of 63
   ipagroup:
     ipaadmin_password: "{{ ipaadmin_password }}"
     name: audio
     idoverrideuser:
     - aduser@addomain.com
     gidnumber: 63
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-aduser-to-audio-group.yml
```

验证

1. 以 AD 用户身份登录 IdM 客户端：

```
$ ssh aduser@addomain.com@client.idm.example.com
```

2. 验证 AD 用户的组成员资格：

```
$ id aduser@addomain.com
uid=702801456(aduser@addomain.com) gid=63(audio) groups=63(audio)
```

其它资源

- idoverrideuser 和 ipagroup ansible-freeipa 上游文档
- 为 IdM 中的本地和远程组启用组合并

## 35.12. 使用 ANSIBLE 确保带有特定 UID 的 ID 视图中存在 IDM 用户

如果您在一个实验室工作，其中您有自己的计算机，但您的 /home/ 目录位于服务器导出的共享驱动器中，您可以有两个用户：

- 一个是系统范围的用户，集中存储在身份管理(IdM)中。
- 其帐户是本地的，该帐户存储在有问题的系统中。

如果您需要完全访问您的文件，无论您是以 IdM 用户或本地用户登录，您可以通过为这两个用户提供相同的 UID 来完成此操作。

完成此流程，使用 ansible-freeipa idoverrideuser 模块：

- 将 ID 视图应用到名为 idview_for_host01 的 host01。
- 在 idview_for_host01 中，确保 idm_user 存在用户 ID 覆盖，其 UID 为 20001。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 您使用 RHEL 8.10 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- idview_for_host1 ID 视图存在。

**流程**

1. 在 Ansible 控制节点上，使用以下内容创建一个 ensure-idmuser-and-local-user-have-access-to-same-files.yml playbook：

```
---
- name: Ensure both local user and IdM user have access to same files
  hosts: ipaserver
  become: false
  gather_facts: false

  tasks:
  - name: Ensure idview_for_host1 is applied to host1.idm.example.com
    ipaidview:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: idview_for_host01
      host: host1.idm.example.com
  - name: Ensure idmuser is present in idview_for_host01 with the UID of 20001
    ipaidoverrideuser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      idview: idview_for_host01
      anchor: idm_user
      UID: 20001
```

2. 保存该文件。

3. 运行 playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory ensure-idmuser-and-local-user-have-access-to-same-files.yml
```

其它资源

- ansible-freeipa 上游文档中的 idoverrideuser 模块

### 35.13. 使用 ANSIBLE 确保 IDM 用户可以使用两个证书登录到 IDM 客户端

如果您希望一个身份管理(IdM)用户通常使用密码登录到 IdM，以便只使用智能卡向特定的 IdM 客户端进行身份验证，您可以创建一个 ID 视图，该视图需要该客户端上的用户认证。

完成此流程，使用 ansible-freeipa idoverrideuser 模块：

- 将 ID 视图应用到名为 idview_for_host01 的 host01。
- 确保在 idview_for_host01 中，为 idm_user 存在带有两个证书的用户 ID 覆盖。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 您使用 RHEL 8.10 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
  - 示例假定 cert1.b64 和 cert2.b64 证书位于您要执行 playbook 的同一目录中。
- idview_for_host01 ID 视图存在。

**流程**

1. 在 Ansible 控制节点上，使用以下内容创建一个 ensure-idmuser-present-in-idview-with-certificates.yml playbook：

```
---
- name: Ensure both local user and IdM user have access to same files
  hosts: ipaserver
  become: false
  gather_facts: false
```

```
tasks:
- name: Ensure idview_for_host1 is applied to host01.idm.example.com
  ipaidview:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: idview_for_host01
    host: host01.idm.example.com

- name: Ensure an IdM user is present in ID view with two certificates
  ipaidoverrideuser:
    ipaadmin_password: "{{ ipaadmin_password }}"
    idview: idview_for_host01
    anchor: idm_user
    certificate:
    - "{{ lookup('file', 'cert1.b64', rstrip=False) }}"
    - "{{ lookup('file', 'cert2.b64', rstrip=False) }}"
```

rstrip=False 指令会导致不会从查找文件末尾删除空格。

2. 保存该文件。

3. 运行 playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory ensure-
idmuser-present-in-idview-with-certificates.yml
```

其它资源

- ansible-freeipa 上游文档中的 idoverrideuser 模块

## 35.14. 使用 ANSIBLE 为 IDM 客户端上的声音卡授予 IDM 组访问权限

您可以使用 ansible-freeipa idview 和 idoverridegroup 模块在 IdM 客户端上使身份管理(IdM)或 Active Directory (AD)用户成员。这会授予 IdM 或 AD 用户对主机上声音卡的特权访问权限。

该流程使用 idview_for_host01 ID 视图的示例，其 音频 组 ID 覆盖使用 GID 的 63 来添加，它对应于 RHEL 主机上本地 音频 组的 GID。idview_for_host01 ID 视图应用于名为 host01.idm.example.com 的 IdM 客户端。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：

- 您使用 Ansible 版本 2.14 或更高版本。
- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 您使用 RHEL 8.10 或更高版本。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

**流程**

1. [可选] 识别 RHEL 主机上本地 音频 组的 GID：

```
$ getent group audio
--------------------
audio:x:63
```

2. 在 Ansible 控制节点上，使用以下任务创建一个 give-idm-group-access-to-sound-card-on-idm-client.yml playbook：

```
---
- name: Playbook to give IdM group access to sound card on IdM client
  hosts: ipaserver
  become: false

  tasks:
  - name: Ensure the audio group exists in IdM
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: audio

  - name: Ensure idview_for_host01 exists and is applied to host01.idm.example.com
    ipaidview:
      ipaadmin_password:  ”{{ ipaadmin_password }}"
      name: idview_for_host01
      host: host01.idm.example.com

  - name: Add an override for the IdM audio group with GID 63 to idview_for_host01
    ipaidoverridegroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
```

```
    idview: idview_for_host01
    anchor: audio
    GID: 63
```

3. 保存该文件。

4. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory give-idm-group-access-to-sound-card-on-idm-client.yml
```

验证

1. 在 IdM 客户端上，获取 IdM 管理员的凭证：

```
$ kinit admin
Password:
```

2. 创建测试 IdM 用户：

```
$ ipa user-add testuser --first test --last user --password
User login [tuser]:
Password:
Enter Password again to verify:
----------------
Added user "tuser"
----------------
```

3. 将用户添加到 IdM 音频组中：

```
$ ipa group-add-member --tuser audio
```

4. 以 tuser 用户身份登录 host01.idm.example.com：

```
$ ssh tuser@host01.idm.example.com
```

5. 验证用户的组成员资格：

```
$ id tuser
uid=702801456(tuser) gid=63(audio) groups=63(audio)
```

其它资源

- idoverridegroup、idview 和 ipagroup ansible-freeipa 上游文档
- 为 IdM 中的本地和远程组启用组合并

### 35.15. 将 NIS 域迁移到身份管理

您可以使用 ID 视图为现有主机设置主机特定的 UID 和 GID，以防止在将 NIS 域迁移到 IdM 时更改文件和目录的权限。

先决条件

- 使用 kinit admin 命令，将自己认证为 admin 。

流程

1. 在 IdM 域中添加用户和组。
   a. 使用 ipa user-add 命令创建用户。如需更多信息，请参阅：将用户添加到 IdM。
   b. 使用 ipa group-add 命令创建组。如需更多信息，请参阅：将组添加到 IdM。
2. 覆盖在用户创建过程中 Idm 生成的 ID：
   a. 使用 ipa idview-add 命令创建一个新的 ID 视图。如需更多信息，请参阅：获取 ID 视图命令的帮助。
   b. 使用 ipa idoverrideuser-add 和 idoverridegroup-add 将用户和组的 ID 覆盖分别添加到 ID 视图。

3. 使用 ipa idview-apply 命令将 ID 视图分配给特定的主机。

4. 停用 NIS 域。

验证

1. 要检查所有用户和组是否已正确添加到 ID 视图中，请使用 ipa idview-show 命令。

```
$ ipa idview-show example-view
  ID View Name: example-view
  User object overrides: example-user1
  Group object overrides: example-group
```

# 第 36 章 为活动目录用户使用 ID 视图

您可以使用 ID 视图为 IdM-AD Trust 环境中活动目录(AD)用户的 POSIX 属性指定新值。

默认情况下，IdM 对所有 AD 用户应用 Default Trust View 。您可以在单个 IdM 客户端上配置其它 ID 视图，以进一步调整特定用户所收到的 POSIX 属性。

## 36.1. DEFAULT TRUST VIEW 是如何工作的

Default Trust View 是默认的 ID 视图，其总是在基于信任的设置被应用于到 AD 用户和组。当您使用 ipa-adtrust-install 命令建立信任时，它会被自动创建，且不能被删除。

**注意**

Default Trust View 仅接受对 AD 用户和组的覆盖，不接受对 IdM 用户和组的覆盖。

使用 Default Trust View，您可以为 AD 用户和组定义自定义 POSIX 属性，从而覆盖 AD 中定义的值。

表 36.1. 应用 Default Trust View

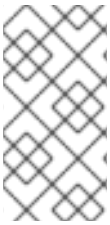
| | AD 中的值 | 默认信任视图 | 结果 |
|---|---|---|---|
| 登录 | ad_user | ad_user | ad_user |
| UID | 111 | 222 | 222 |
| GID | 111 | （无值） | 111 |

您还可以配置其它 ID 视图来覆盖 IdM 客户端上的 Default Trust View。IdM 在 Default Trust View 顶部应用特定于主机的 ID 视图中的值：

- 如果特定于主机的 ID 视图中定义了一个属性，则 IdM 会应用此 ID 视图中的值。
- 如果在特定于主机的 ID 视图中未定义一个属性，则 IdM 会应用 Default Trust View 中的值。

表 36.2. 在 Default Trust View 顶部应用特定于主机的 ID 视图

| | AD 中的值 | 默认信任视图 | 特定主机的 ID 视图 | 结果 |
|---|---|---|---|---|
| 登录 | ad_user | ad_user | （无值） | ad_user |
| UID | 111 | 222 | 333 | 333 |
| GID | 111 | （无值） | 333 | 333 |



**注意**

您只能应用特定于主机的 ID 视图来覆盖 IdM 客户端上的 Default Trust View。IdM 服务器和副本总是应用 Default Trust View 中的值。

**其它资源**

- 使用 ID 视图来覆盖 IdM 客户端上的用户属性值

## 36.2. 通过修改 DEFAULT TRUST VIEW 为 AD 用户定义全局属性

如果要在整个 IdM 部署中覆盖活动目录(AD)用户的 POSIX 属性，请在 Default Trust View 中修改该用户的条目。这个过程将 AD 用户 ad_user@ad.example.com 的 GID 设为 732000006。

**先决条件**

- 您已作为 IdM 管理员进行身份验证。
- 具有 GID 的组必须存在，否则您必须在组的 ID 覆盖中设置 GID。

**流程**

1. 作为 IdM 管理员，在 Default Trust View 中为 AD 用户创建一个 ID 覆盖，其将 GID 号更改为 732000006：

```
# ipa idoverrideuser-add 'Default Trust View' ad_user@ad.example.com --gidnumber=732000006
```

2. 从所有 IdM 服务器和客户端上的 SSSD 缓存中清除 ad_user@ad.example.com 用户的条

目。这会删除过时的数据，并允许应用新的覆盖值。

```
# sssctl cache-expire -u ad_user@ad.example.com
```

验证

- 检索 ad_user@ad.example.com 用户的信息以验证 GID 是否反映了更新的值。

```
# id ad_user@ad.example.com
uid=702801456(ad_user@ad.example.com) gid=732000006(ad_admins)
groups=732000006(ad_admins),702800513(domain users@ad.example.com)
```

## 36.3. 对带有 ID 视图的 IDM 客户端上的 AD 用户覆盖 DEFAULT TRUST VIEW 属性

您可能希望为活动目录(AD)用户覆盖 Default Trust View 中的一些 POSIX 属性。例如，您可能需要在一个特定的 IdM 客户端上给 AD 用户赋予一个不同的 GID。对 AD 用户，您可以使用一个 ID 视图覆盖 Default Trust View 中的一个值，并将其应用到单个主机。此流程解释了如何将 host1.idm.example.com IdM 客户端上的 ad_user@ad.example.com AD 用户的 GID 设为 732001337。

先决条件

- 您有访问 host1.idm.example.com IdM 客户端的 root 权限。
- 您已作为具有所需权限的用户登录了，如 admin 用户。

流程

1. 创建 ID 视图。例如，创建名为 example_for_host1 的 ID 视图：

```
$ ipa idview-add example_for_host1
---------------------------
Added ID View "example_for_host1"
---------------------------
  ID View Name: example_for_host1
```

2. 将用户覆盖添加到 example_for_host1 ID 视图。要覆盖用户的 GID：

    - 输入 ipa idoverrideuser-add 命令

- 添加 ID 视图的名称
- 添加用户名，也称为锚
- 添加 --gidnumber= 选项：

```
$ ipa idoverrideuser-add example_for_host1 ad_user@ad.example.com --
gidnumber=732001337
----------------------------
Added User ID override "ad_user@ad.example.com"
----------------------------
  Anchor to override: ad_user@ad.example.com
  GID: 732001337
```

3. 将 example_for_host1 应用到 host1.idm.example.com IdM 客户端：

```
$ ipa idview-apply example_for_host1 --hosts=host1.idm.example.com
----------------------------
Applied ID View "example_for_host1"
----------------------------
hosts: host1.idm.example.com
------------------------------------------
Number of hosts the ID View was applied to: 1
------------------------------------------
```

**注意**

ipa idview-apply 命令也接受 --hostgroups 选项。选项将 ID 视图应用到属于指定主机组的主机，但不会将 ID 视图与主机组本身相关联。相反，--hostgroups 选项会展开指定主机组的成员，并将 --hosts 选项分别应用到其中的每一个成员。

这意味着，如果以后将主机添加到主机组中，则 ID 视图不会应用到新主机。

4. 从 host1.idm.example.com IdM 客户端上的 SSSD 缓存中清除掉 ad_user@ad.example.com 用户的条目。这会删除过时的数据，并允许应用新的覆盖值。

```
[root@host1 ~]# sssctl cache-expire -u ad_user@ad.example.com
```

验证步骤

1. 以 ad_user@ad.example.com 身份 SSH 到 host1 ：

```
[root@r8server ~]# ssh ad_user@ad.example.com@host1.idm.example.com
```

2. 检索 ad_user@ad.example.com 用户的信息以验证 GID 是否反映了更新的值。

```
[ad_user@ad.example.com@host1 ~]$ id ad_user@ad.example.com
uid=702801456(ad_user@ad.example.com) gid=732001337(admins2)
groups=732001337(admins2),702800513(domain users@ad.example.com)
```

## 36.4. 将 ID 视图应用到 IDM 主机组

ipa idview-apply 命令接受 --hostgroups 选项。不过，选项充当一次性操作，它将 ID 视图应用到当前属于指定主机组的主机，但不动态地将 ID 视图与主机组本身关联。--hostgroups 选项将展开指定主机组的成员，并将 --hosts 选项分别应用到其中的每一个成员。

如果稍后向主机组添加新主机，您必须使用 ipa idview-apply 命令及 --hosts 选项，手动将 ID 视图应用到新主机。

类似地，如果您从主机组中删除主机，则移除后 ID 视图仍会分配给该主机。要从删除的主机中取消 ID 视图应用，您必须运行 ipa idview-unapply *id_view_name* --hosts=*name_of_the_removed_host* 命令。

按照以下流程实现以下目标：

1. 如何创建主机组并向其添加主机。
2. 如何将 ID 视图应用到主机组。
3. 如何向主机组添加新主机，并将 ID 视图应用到新主机。

**先决条件**

- 确保 IdM 中存在您要应用到主机组的 ID 视图。例如：要创建一个 ID 视图来覆盖特定 IdM 客户端上的 IdM 用户登录名称，请参阅 使用 ID 视图覆盖特定主机上 IdM 用户的登录名称。

**流程**

1. 创建主机组并为其添加主机：

   a. 创建主机组.例如，创建名为 baltimore 的主机组：

   ```
   [root@server ~]# ipa hostgroup-add --desc="Baltimore hosts" baltimore
   ----------------------------
   Added hostgroup "baltimore"
   ----------------------------
   Host-group: baltimore
   Description: Baltimore hosts
   ```

   b. 将主机添加到主机组。例如，将 host102 和 host103 添加到 baltimore 主机组：

   ```
   [root@server ~]# ipa hostgroup-add-member --hosts={host102,host103} baltimore
   Host-group: baltimore
   Description: Baltimore hosts
   Member hosts: host102.idm.example.com, host103.idm.example.com
   -------------------------
   Number of members added 2
   -------------------------
   ```

2. 将 ID 视图应用到主机组中的主机。例如，要将 example_for_host1 ID 视图应用到 baltimore 主机组：

   ```
   [root@server ~]# ipa idview-apply --hostgroups=baltimore
   ID View Name: example_for_host1
   -----------------------------------------
   Applied ID View "example_for_host1"
   -----------------------------------------
     hosts: host102.idm.example.com, host103.idm.example.com
   ------------------------------------------------
   Number of hosts the ID View was applied to: 2
   ------------------------------------------------
   ```

3. 将新主机添加到主机组，并将 ID 视图应用到新主机：

   a. 将新主机添加到主机组。例如，要将 somehost.idm.example.com 主机添加到 baltimore 主机组：

   ```
   [root@server ~]# ipa hostgroup-add-member --hosts=somehost.idm.example.com
   baltimore
   ```

```
  Host-group: baltimore
  Description: Baltimore hosts
  Member hosts:  host102.idm.example.com,
host103.idm.example.com,somehost.idm.example.com
-------------------------
Number of members added 1
-------------------------
```

b. （可选）显示 ID 视图信息。例如，要显示 example_for_host1 ID 视图的详情：

```
[root@server ~]# ipa idview-show example_for_host1 --all
  dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
  ID View Name: example_for_host1
[...]
  Hosts the view applies to: host102.idm.example.com, host103.idm.example.com
  objectclass: ipaIDView, top, nsContainer
```

输出显示 ID 视图没有应用到 somehost.idm.example.com（在 baltimore 主机组中新添加的主机）。

c. 将 ID 视图应用到新主机。例如，要将 example_for_host1 ID 视图应用到 somehost.idm.example.com ：

```
[root@server ~]# ipa idview-apply --host=somehost.idm.example.com
ID View Name: example_for_host1
-----------------------------------------
Applied ID View "example_for_host1"
-----------------------------------------
  hosts: somehost.idm.example.com
-------------------------------------------
Number of hosts the ID View was applied to: 1
-------------------------------------------
```

验证步骤

- 再次显示 ID 视图信息：

```
[root@server ~]# ipa idview-show example_for_host1 --all
  dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
  ID View Name: example_for_host1
[...]
  Hosts the view applies to: host102.idm.example.com, host103.idm.example.com,
somehost.idm.example.com
  objectclass: ipaIDView, top, nsContainer
```

输出显示 ID 视图现在已应用到 somehost.idm.example.com（在 baltimore 主机组中新添

**加的主机）。**

# 第 37 章 手动调整 ID 范围

IdM 服务器生成唯一用户 ID(UID)和组 ID(GID)号。通过为副本创建和分配不同的 ID 范围，还确保它们永远不会生成相同的 ID 号。默认情况下，此过程是自动的。但是，您可以在 IdM 服务器安装过程中手动调整 IdM ID 范围，或者手动定义副本的 DNA ID 范围。

## 37.1. ID 范围

ID 号被划分为 *ID 范围*。为各个服务器和副本保持单独的数字范围可避免为某个条目发布的 ID 号已在其他服务器或副本上的另一个条目使用的几率。

请注意，有两种不同的 ID 范围：

- IdM *ID 范围*，是在安装第一个服务器时分配的。此范围在创建后不可修改。但是，除了新的 IdM ID 范围外，您还可以创建新的 IdM ID 范围。如需更多信息，请参阅 自动 ID 范围分配 和 添加一个新的 IdM ID 范围。

- *分布式数字分配*（DNA）ID 范围，可由用户修改。它们必须适合现有的 IdM ID 范围。如需更多信息，请参阅 手动分配 DNA ID 范围。

  也可以给副本分配 下一个 DNA ID 范围。当副本当前范围内的 ID 不足时，副本会使用其下一个范围。当一个副本被删除时，下一个范围不会被 自动分配，因此您必须手动分配它们。

作为域的后端 389 目录服务器实例的一部分，范围是通过 DNA 插件在服务器和副本之间更新和共享的。

DNA 范围定义由两个属性设置：

- 服务器的下一个可用数字： DNA 范围的低端

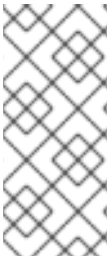
- 范围大小：ID 在 DNA 范围内的数量

初始底部范围是在插件实例配置期间设置的。之后，插件会更新底部值。通过将可用号划分成范围，服务器可以持续分配号，而不会相互重叠。

## 37.2. 自动 ID 范围分配

### IdM ID 范围

默认情况下，IdM ID 范围会在 IdM 服务器安装过程中自动分配。ipa-server-install 命令会从总共 10,000 个可能的范围中随机选择并分配 200,000 个 ID。当您决定以后合并两个独立的 IdM 域时，以这种方法选择一个随机范围可显著降低冲突 ID 的可能性。



**注意**

此 IdM ID 范围在创建后不能修改。您只能手动使用 分配 DNA ID 范围中介绍的命令手动调整分布式 Numeric Assignment(DNA)ID 范围。与 IdM ID 范围匹配的 DNA 范围是在安装过程中自动创建的。

### DNA ID 范围

如果您安装了一个 IdM 服务器，它会控制整个 DNA ID 范围。当您安装了新副本，并且副本请求它自己的 DNA ID 范围时，服务器的初始 ID 范围将被拆分，并分布在服务器和副本之间：副本接收初始服务器上可用的剩余 DNA ID 范围的一半。服务器和副本随后将原始 ID 范围的各自部分用于新用户或组条目。另外，如果副本即将耗尽其分配的 ID 范围，且剩余的ID 少于 100 个，则副本会联系其他可用的服务器来请求新的 DNA ID 范围。

**重要**

安装副本时，它不会 立即收到一个 ID 范围。副本在首次使用 DNA 插件时收到一个 ID 范围，例如首次添加用户时。

如果初始服务器在副本向其请求 DNA ID 范围之前停止工作，则副本无法与服务器联系来请求 ID 范围。尝试在副本上添加新用户会失败。在这种情况下，您可以找出分配给禁用的服务器的 ID 范围，并 手动为副本分配一个 ID 范围。

## 37.3. 在服务器安装过程中手动分配 IDM ID 范围

您可以覆盖默认行为，并手动设置 IdM ID 范围，而不是随机分配。

**重要**

不要设置 UID 值为 1000 或更低的 ID 范围；这些值是保留给系统使用的。另外，不要设置包含 0 值的 ID 范围；SSSD 服务不处理 ID 为 0 的值。

流程

- 您可以在服务器安装过程中使用 ipa-server-install 及以下两个选项来手动定义 IdM ID 范围：
  - --idstart 给出 UID 和 GID 号的起始值。
  - --idmax 给出 UID 和 GID 号的最大值；默认情况下，值为 --idstart 起始值加上 199,999。

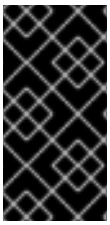
验证步骤

- 要检查 ID 范围是否已正确分配，您可以使用 ipa idrange-find 命令显示已分配的 IdM ID 范围：

```
# ipa idrange-find
----------------
1 range matched
----------------
  Range name: IDM.EXAMPLE.COM_id_range
  First Posix ID of the range: 882200000
  Number of IDs in the range: 200000
  Range type: local domain range
----------------------------
Number of entries returned 1
----------------------------
```

## 37.4. 添加新的 IDM ID 范围

在某些情况下，除了原始的 ID 范围，您可能想要创建新的 IdM ID 范围；例如，当副本的 ID 用完，且原始的 IdM ID 范围耗尽时。



重要

添加新 IdM ID 范围不会自动创建新的 DNA ID 范围。您必须根据需要手动将新的 DNA ID 范围分配给副本。有关如何进行此操作的更多信息，请参阅 手动分配 DNA ID 范围。

流程

1. 要创建新的 IdM ID 范围，请使用 ipa idrange-add 命令。您必须指定新范围名称、范围的第一个 ID 号和范围大小：

```
# ipa idrange-add IDM.EXAMPLE.COM_new_range --base-id=1000000 --range-
size=200000
----------------------------------------
Added ID range "IDM.EXAMPLE.COM_new_range"
----------------------------------------
  Range name: IDM.EXAMPLE.COM_new_range
  First Posix ID of the range: 1000000
  Number of IDs in the range: 200000
  Range type: local domain range
```

2. 重启 Directory 服务器：

```
# systemctl restart dirsrv@IDM.EXAMPLE.COM.service
```

这确保当您使用新范围中的 UID 创建用户时，它们分配了安全标识符(SID)。

3. 可选：立即更新 ID 范围：

   a. 清除系统安全服务守护进程(SSSD)缓存：

   ```
   # sss_cache -E
   ```

   b. 重启 SSSD 守护进程：

   ```
   # systemctl restart sssd
   ```

**注意**

如果您没有清除 SSSD 缓存并重启服务，SSSD 仅在更新域列表和其他存储在 IdM 服务器中的配置数据时检测到新的 ID 范围。

验证步骤

- 您可以使用 ipa idrange-find 命令检查新范围是否设置正确：

```
# ipa idrange-find
----------------
2 ranges matched
----------------
  Range name: IDM.EXAMPLE.COM_id_range
```

```
  First Posix ID of the range: 882200000
  Number of IDs in the range: 200000
  Range type: local domain range

  Range name: IDM.EXAMPLE.COM_new_range
  First Posix ID of the range: 1000000
  Number of IDs in the range: 200000
  Range type: local domain range
----------------------------
Number of entries returned 2
----------------------------
```

## 37.5. IDM ID 范围内安全和相对标识符的角色

Identity Management(IdM)ID 范围由多个参数定义：

- 范围名称
- 范围的第一个 POSIX ID
- 范围大小： 范围内的 ID 数量
- 对应的 RID range 的第一个 相对标识符 (RID)
- 二级 RID 范围的第一个 RID

您可以使用 ipa idrange-show 命令查看这些值：

```
$ ipa idrange-show IDM.EXAMPLE.COM_id_range
  Range name: IDM.EXAMPLE.COM_id_range
  First Posix ID of the range: 196600000
  Number of IDs in the range: 200000
  First RID of the corresponding RID range: 1000
  First RID of the secondary RID range: 1000000
  Range type: local domain range
```

安全识别符

IdM 服务器在内部使用本地域的 ID 范围中的数据，来将唯一的 安全标识符 (SIDs)分配给 IdM 用户和组。SID 存储在用户和组对象中。用户的 SID 由以下内容组成：

- 域 SID

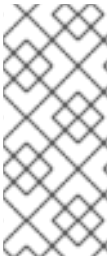
- 用户的 相对标识符 (RID)，它是附加到域 SID 的四位 32 位值

例如，如果域 SID 是 S-1-5-21-123-456-789，且来自此域的 RID 为 1008，则该用户的 SID 为 S-1-5-21-123-456-789-1008。

相对标识符

RID 本身以以下方式计算：

将用户的 POSIX UID 范围内的第一个 POSIX ID 减去，并将对应 RID 范围的第一个 RID 添加到结果中。例如，如果 *idmuser* 的 UID 为 196600008，则第一个 POSIX ID 是 196600000，第一个 RID 为 1000，则 *idmuser* 的 RID 为 1008。



注意

该算法计算用户的 RID 会检查给定的 POSIX ID 是否位于计算对应的 RID 前分配的 ID 范围内。例如，如果第一个 ID 是 196600000，其范围大小为 200000，则 1600000 的 POSIX ID 不在 ID 范围之外，且算法不会为其计算 RID。

二级相对标识符

在 IdM 中，POSIX UID 可以与 POSIX GID 相同。这意味着，如果 *idmuser* 已被 UID 196600008 存在，您仍然可以创建一个新的 *idmgroup* 组，其 GID 为 196600008。

但是，SID 只能定义一个对象、*用户或组群*。已经为 *idmuser* 创建的 SID 是 S-1-5-21-123-456-789-1008，它无法与 *idmgroup* 共享。必须为 *idmgroup* 生成一个替代 SID。

IdM 使用 二级相对标识符，或辅助 RID 来避免冲突 SID。这个二级 RID 由以下内容组成：

- secondary RID 基本

- 范围大小；默认情况下，与基本范围大小相同

在上例中，二级 RID 基本被设置为 1000000。要计算新创建的 *idmgroup* 的 RID：从用户的 POSIX UID 中减去范围内的第一个 POSIX ID，并将 secondary RID 范围内的第一个 RID 添加到结果。因此，Mid *group* 被分配为 1000008 的 RID。因此，*idmgroup* 的 SID 是 S-1-5-21-123-456-789-1000008。

只有之前使用手动设置 POSIX ID 创建一个组对象时，IdM 才使用二级 RID 来计算 SID。否则，自动分配可防止分配相同的 ID 两次。

其它资源

- 使用 Ansible 添加新的本地 IdM ID 范围

## 37.6. 使用 ANSIBLE 添加新的本地 IDM ID 范围

在某些情况下，您可能需要创建新的 Identity Management (IdM) ID 范围以及原始的 ID 范围；例如，当副本退出 ID 且原始 IdM ID 范围相同时，原始 IdM ID 范围会被处理。以下示例演示了如何使用 Ansible playbook 创建新 IdM ID 范围。

注意

添加新 IdM ID 范围不会自动创建新的 DNA ID 范围。您需要根据需要手动分配新的 DNA ID 范围。有关如何进行此操作的更多信息，请参阅 手动分配 DNA ID 范围。

先决条件

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。

- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 使用以下内容创建 idrange-present.yml playbook：

```
---
- name: Playbook to manage idrange
  hosts: ipaserver
  become: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure local idrange is present
    ipaidrange:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: new_id_range
      base_id: 12000000
      range_size: 200000
      rid_base: 1000000
      secondary_rid_base: 200000000
```

3. 保存该文件。

4. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory idrange-present.yml
```

5. SSH 到 ipaserver ,并重启 Directory 服务器：

```
# systemctl restart dirsrv@IDM.EXAMPLE.COM.service
```

这确保当您使用新范围中的 UID 创建用户时，它们分配了安全标识符(SID)。

6. 可选：立即更新 ID 范围：

   a. 在 ipaserver 上，清除系统安全服务守护进程(SSSD)缓存：

   ```
   # sss_cache -E
   ```

   b. 在 ipaserver 上，重启 SSSD 守护进程：

   ```
   # systemctl restart sssd
   ```

**注意**

如果您没有清除 SSSD 缓存并重启服务，SSSD 仅在更新域列表和其他存储在 IdM 服务器中的配置数据时检测到新的 ID 范围。

验证步骤

- 您可以使用 ipa idrange-find 命令检查新范围是否设置正确：

```
# ipa idrange-find
----------------
2 ranges matched
----------------
  Range name: IDM.EXAMPLE.COM_id_range
  First Posix ID of the range: 882200000
  Number of IDs in the range: 200000
  Range type: local domain range

  Range name: IDM.EXAMPLE.COM_new_id_range
  First Posix ID of the range: 12000000
  Number of IDs in the range: 200000
  Range type: local domain range
```

```
----------------------------
Number of entries returned 2
----------------------------
```

**其它资源**

- IdM ID 范围中的安全性和相对标识符的角色

## 37.7. 删除对 AD 的信任后删除 ID 范围

如果您已删除了 IdM 和活动目录(AD)环境之间的信任，则您可能想要删除与其关联的 ID 范围。

**警告**

分配给与可信域相关联的 ID 范围的 ID，可能仍然用于注册到 IdM 的系统上的文件和目录的所有权。

如果您删除了与已删除的 AD 信任对应的 ID 范围，则您将无法解析 AD 用户所拥有的任何文件和目录的所有权。

**先决条件**

- 您已删除了对 AD 环境的信任。

**流程**

1. 显示所有当前正在使用的 ID 范围：

```
[root@server ~]# ipa idrange-find
```

2. 识别与您删除的信任相关联的 ID 范围的名称。ID 范围名称的第一部分是信任的名称，如 AD.EXAMPLE.COM_id_range。

3. 删除范围：

```
[root@server ~]# ipa idrange-del AD.EXAMPLE.COM_id_range
```

4. 重启 SSSD 服务，来删除对您已删除的 ID 范围的引用。

```
[root@server ~]# systemctl restart sssd
```

其它资源

- 请参阅 使用命令行删除信任。
- 请参阅 使用 IdM Web UI 删除信任。

## 37.8. 显示当前分配的 DNA ID 范围

您可以显示服务器上当前活跃的分布式数字分配(DNA)ID 范围，以及它的下一个 DNA 范围(如果已经分配了一个)。

流程

- 要显示拓扑中为服务器配置了哪些 DNA ID 范围，请使用以下命令：
  - ipa-replica-manage dnarange-show 显示当前在所有服务器上设置的 DNA ID 范围；或者，如果您指定了一个服务器，则仅显示指定服务器上的 DNA ID 范围，例如：

```
# ipa-replica-manage dnarange-show
serverA.example.com: 1001-1500
serverB.example.com: 1501-2000
serverC.example.com: No range set

# ipa-replica-manage dnarange-show serverA.example.com
serverA.example.com: 1001-1500
```

  - ipa-replica-manage dnanextrange-show 显示当前在所有服务器上设置的下一个 DNA ID 范围；或者，如果您指定了一个服务器，则仅显示指定服务器上的下一个 DNA ID 范围，例如：

```
# ipa-replica-manage dnanextrange-show
serverA.example.com: 2001-2500
serverB.example.com: No on-deck range set
serverC.example.com: No on-deck range set

# ipa-replica-manage dnanextrange-show serverA.example.com
serverA.example.com: 2001-2500
```

## 37.9. 手动 ID 范围分配

在某些情况下，需要手动分配分布式 Numeric Assignment(DNA)ID 范围，例如：

- 副本的 ID 不足，并且 IdM ID 范围已耗尽

  副本已耗尽为其分配的 DNA ID 范围，并且请求额外 ID 失败，因为 IdM 范围内没有更多可用的 ID。

  要解决这种情况，请扩展分配给副本的 DNA ID 范围。您可以通过两种方式执行此操作：

  - 缩短分配给不同副本的 DNA ID 范围，然后将新的可用值分配给已耗尽的副本。
  - 创建新的 IdM ID 范围，然后在这个创建的 IdM 范围内为副本设置一个新的 DNA ID 范围。

    有关如何创建新 IdM ID 范围的详情，请参考 添加一个新的 IdM ID 范围。

- 副本停止工作

  当副本停止运行且必须被删除时，副本的 DNA ID 范围不会被自动检索，这意味着之前分配给副本的 DNA ID 范围不可用。您要恢复 DNA ID 范围，并使其可用于其他副本。

  为此，请在手动 将该范围值分配给不同的服务器前了解 ID 范围值是什么。此外，为了避免重复的 UID 或 GID，请确保恢复范围内的 ID 值之前没有分配给用户或组；您可以通过检查现有用户和组的 UID 和 GID 来完成此操作。

您可以使用手动分配 DNA ID 范围中的命令手动将 DNA ID 范围分配给 副本。

注意

如果您分配了新的 DNA ID 范围，则服务器或副本上已存在条目的 UID 保持不变。这不会造成问题，因为即使您更改了当前的 DNA ID 范围，IdM 也会保留过去分配的范围的记录。

## 37.10. 手动分配 DNA ID 范围

在某些情况下，您可能需要手动将分布式 Numeric Assignment(DNA)ID 范围分配给现有副本，例如，将 DNA ID 范围重新分配给一个无法正常工作的副本。如需更多信息，请参阅 手动 ID 范围分配。

在手动调整 DNA ID 范围时，请确保新调整的范围包含在 IdM ID 范围内；您可以使用 ipa idrange-find 命令对此进行检查。否则，命令会失败。

重要

注意不要创建重叠的 ID 范围。如果您分配给服务器或副本的任何 ID 范围重叠了，可能会导致两个不同的服务器给不同的条目分配了相同的 ID 值。

**先决条件**

- *可选*。如果您要从不工作的副本恢复 DNA ID 范围，首先使用 显示当前分配的 DNA ID 范围 中描述的命令来查找 ID 范围。

**流程**

- 要为指定服务器定义当前的 DNA ID 范围，请使用 ipa-replica-managerangearange-set ：

```
# ipa-replica-manage dnarange-set serverA.example.com 1250-1499
```

- 要为指定的服务器定义下一个 DNA ID 范围，请使用 ipa-replica-manage DNanextrange-set ：

```
# ipa-replica-manage dnanextrange-set serverB.example.com 1500-5000
```

**验证步骤**

- 您可以使用 显示当前分配的 DNA ID 范围 中描述的命令来检查新的 DNA 范围是否设置正

确。

# 第 38 章 手动管理 SUBID 范围

在容器化环境中，有时 IdM 用户需要手动分配 subID 范围。以下说明描述了如何管理 subID 范围。

## 38.1. 使用 IDM CLI 生成子 SUBID 范围

作为身份管理(IdM)管理员，您可以生成一个 subID 范围，并将其分配给 IdM 用户。

**先决条件**

- IdM 用户存在。
- 已获得 IdM admin ticket-granting ticket (TGT)。如需了解更多详细信息，请参阅使用 kinit 手动登录到 IdM。
- 您有访问您要执行该流程的 IdM 主机的 root 访问权限。

**流程**

1. [可选] 检查现有的 subID 范围：

```
# ipa subid-find
```

2. 如果 subID 范围不存在，请选择以下选项之一：
   - 生成并为 IdM 用户分配 subID 范围：

```
# ipa subid-generate --owner=idmuser

Added subordinate id "359dfcef-6b76-4911-bd37-bb5b66b8c418"

  Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
  Description: auto-assigned subid
  Owner: idmuser
  SubUID range start: 2147483648
  SubUID range size: 65536
  SubGID range start: 2147483648
  SubGID range size: 65536
```

- 生成并分配 subID 范围到所有 IdM 用户：

```
# /usr/libexec/ipa/ipa-subids --all-users

Found 2 user(s) without subordinate ids
  Processing user 'user4' (1/2)
  Processing user 'user5' (2/2)
Updated 2 user(s)
The ipa-subids command was successful
```

3. [可选] 默认将 subID 范围分配给新的 IdM 用户：

```
# ipa config-mod --user-default-subid=True
```

验证

- 验证用户是否已分配 subID 范围：

```
# ipa subid-find --owner=idmuser

1 subordinate id matched

  Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
  Owner: idmuser
  SubUID range start: 2147483648
  SubUID range size: 65536
  SubGID range start: 2147483648
  SubGID range size: 65536

Number of entries returned 1
```

## 38.2. 使用 IDM WEBUI 接口生成 SUBID 范围

作为身份管理(IdM)管理员，您可以生成一个 subID 范围，并在 IdM WebUI 界面中将其分配给用户。

先决条件

- IdM 用户存在。
- 您已获得 IdM admin Kerberos 票据(TGT)。请参阅 在 Web UI 中登录到 IdM：使用 Kerberos 票据 以了解更多详细信息。

- 您有访问您要执行该流程的 IdM 主机的 root 访问权限。

流程

1. 在 IdM WebUI 界面中，展开 Subordinate ID 选项卡，然后选择 Subordinate ID 选项。
2. 当显示 Subordinate ID 接口时，点界面右上角的 Add 按钮。此时会出现 Add subid 窗口。
3. 在 Add subid 窗口中，选择一个所有者，这是您要为其分配 subID 范围的用户。
4. 点击 Add 按钮。

验证

- 查看 Subordinate IDs 选项卡下的表。表中显示了一条新记录。所有者是您为其分配 subID 范围的用户。

### 38.3. 使用 IDM CLI 查看有关 IDM 用户的 SUBID 信息

作为身份管理(IdM)用户，您可以搜索 IdM 用户 subID 范围并查看相关信息。

先决条件

- 您已在 IdM 客户端 中配置了 subID 范围。
- 您已获得 IdM 用户票据授予票(TGT)。如需了解更多详细信息，请参阅使用 kinit 手动登录到 IdM。

流程

- 查看 subID 范围的详情：
  - 如果您知道是范围所有者的 Identity Management (IdM)用户的唯一 ID 哈希：

```
$ ipa subid-show 359dfcef-6b76-4911-bd37-bb5b66b8c418

  Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
  Owner: idmuser
  SubUID range start: 2147483648
  SubUID range size: 65536
  SubGID range start: 2147483648
  SubGID range size: 65536
```

- 如果您知道该范围内的特定 subID：

```
$ ipa subid-match --subuid=2147483670

1 subordinate id matched

  Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
  Owner: uid=idmuser
  SubUID range start: 2147483648
  SubUID range size: 65536
  SubGID range start: 2147483648
  SubGID range size: 65536

Number of entries returned 1
```

## 38.4. 使用 GETSUBID 命令列出 SUBID 范围

作为系统管理员，您可以使用命令行界面列出身份管理(IdM)或本地用户的 subID 范围。

**先决条件**

- idmuser 用户存在于 IdM 中。
- 已安装 shadow-utils-subid 软件包。
- 您可以编辑 /etc/nsswitch.conf 文件。

**流程**

1. 打开 /etc/nsswitch.conf 文件，并通过将 subid 变量设置为 sss 值将 shadow-utils 工具配置为使用 IdM subID 范围：

```
[...]
subid: sss
```

**注意**

您只能为 subid 字段提供一个值。将 subid 字段设置为 file 值或 no 值，而不是 sss 将 shadow-utils 工具配置为使用 /etc/subuid 和 /etc/subgid 文件中的 subID 范围。

2. 列出 IdM 用户的 subID 范围：

```
$ getsubids idmuser
0: idmuser 2147483648 65536
```

第一个值 2147483648 表示 subID 范围 start。第二个值 65536 表示范围的大小。

# 第 39 章 使用 ANSIBLE 管理 IDM 中的复制拓扑

您可以维护多个身份管理 (IdM) 服务器，并使它们相互复制，以实现冗余目的，以减少或防止服务器丢失。例如，如果一个服务器失败，其他服务器就会为域提供服务。您还可以根据剩余的服务器创建新副本来恢复丢失的服务器。

存储在 IdM 服务器上的数据会根据复制协议复制：当两台服务器配置了复制协议时，它们将共享其数据。复制的数据存储在拓扑后缀中。当两个副本在其后缀之间具有复制协议时，后缀组成一个拓扑片段（segment）。

本章论述了如何使用 Red Hat Ansible Engine 管理 IdM 复制协议、拓扑片段和拓扑后缀。本章包含以下部分：

- 使用 Ansible 确保 IdM 中存在复制协议
- 使用 Ansible 确保多个 IdM 副本之间存在复制协议
- 使用 Ansible 检查两个副本之间是否存在复制协议
- 使用 Ansible 验证 IdM 中是否存在拓扑后缀
- 使用 Ansible 重新初始化 IdM 副本
- 使用 Ansible 确保 IdM 中没有复制协议

## 39.1. 使用 ANSIBLE 确保 IDM 中存在复制协议

存储在身份管理 (IdM) 服务器上的数据存储基于复制协议：配置了两个服务器时，它们共享其数据。复制协议始终为现实：数据从第一个副本复制到另一个副本，另一个副本复制到第一个副本。

按照以下流程，使用 Ansible playbook 确保 server.idm.example.com 和 replica.idm.example.com 之间存在 domain 类型的复制协议。

**先决条件**

- 确保您了解 连接拓扑中 IdM 副本指南 中列出的 设计 IdM 拓扑 的建议。
- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/topology/ 目录中的 add-topologysegment.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegment.yml
add-topologysegment-copy.yml
```

3. 打开 add-topologysegment-copy.yml 文件进行编辑。

4. 通过在 ipatopologysegment 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM admin 的密码。
- 根据您要添加的分段类型，将 suffix 变量设置为 domain 或 ca。
- 将 left 变量设置为您要作为复制协议左侧节点的 IdM 服务器的名称。
- 将 right 变量设置为您要作为复制协议正确节点的 IdM 服务器的名称。
- 确保 state 变量设置为 present。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to handle topologysegment
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
- name: Add topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      left: server.idm.example.com
      right: replica.idm.example.com
      state: present
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-
topologysegment-copy.yml
```

**其它资源**

- 请参阅 解释复制协议、拓扑后缀和拓扑段。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-topology.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/topology 目录中的 playbook 示例。

## 39.2. 使用 ANSIBLE 确保多个 IDM 副本之间存在复制协议

存储在身份管理 (IdM) 服务器上的数据存储基于复制协议：配置了两个服务器时，它们共享其数据。复制协议始终为现实：数据从第一个副本复制到另一个副本，另一个副本复制到第一个副本。

按照以下流程确保复制协议在 IdM 中的多个副本对之间存在。

**先决条件**

- 确保您了解拓扑中连接副本中列出的 IdM 拓扑的建议。
- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
-

目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/topology/ 目录中的 add-topologysegments.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegments.yml
add-topologysegments-copy.yml
```

3. 打开 add-topologysegments-copy.yml 文件进行编辑。

4. 通过在 vars 部分中设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM admin 的密码。
- 对于每个拓扑片段，在 ipatopology_segments 部分添加一个行并设置以下变量：
  - 根据您要添加的分段类型，将 suffix 变量设置为 domain 或 ca。
  - 将 left 变量设置为您要作为复制协议左侧节点的 IdM 服务器的名称。
  - 将 right 变量设置为您要作为复制协议正确节点的 IdM 服务器的名称。

5. 在 add-topologysegments-copy.yml 文件的 tasks 部分中，确保 state 变量设置为 present。

这是当前示例修改的 Ansible playbook 文件：

-

```
---
- name: Add topology segments
  hosts: ipaserver
  gather_facts: false

  vars:
    ipaadmin_password: "{{ ipaadmin_password }}"
    ipatopology_segments:
    - {suffix: domain, left: replica1.idm.example.com , right: replica2.idm.example.com }
    - {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
    - {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
    - {suffix: domain+ca, left: replica4.idm.example.com , right:
replica1.idm.example.com }

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Add topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: "{{ item.suffix }}"
      name: "{{ item.name | default(omit) }}"
      left: "{{ item.left }}"
      right: "{{ item.right }}"
      state: present
      #state: absent
      #state: checked
      #state: reinitialized
    loop: "{{ ipatopology_segments | default([]) }}"
```

6. 保存该文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-
topologysegments-copy.yml
```

**其它资源**

- 请参阅 解释复制协议、拓扑后缀和拓扑段。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-topology.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/topology 目录中的 playbook 示例。

## 39.3. 使用 ANSIBLE 检查两个副本之间是否存在复制协议

存储在身份管理 (IdM) 服务器上的数据存储基于复制协议：配置了两个服务器时，它们共享其数据。复制协议始终为现实：数据从第一个副本复制到另一个副本，另一个副本复制到第一个副本。

按照以下流程验证是否复制协议在 IdM 中的多个副本对之间存在。

### 先决条件

- 确保您了解拓扑中连接副本中列出的 IdM 拓扑的建议。
- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

### 流程

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2.

复制位于 /usr/share/doc/ansible-freeipa/playbooks/topology/ 目录中的 check-topologysegments.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/check-topologysegments.yml
check-topologysegments-copy.yml
```

3. 打开 check-topologysegments-copy.yml 文件进行编辑。

4. 通过在 vars 部分中设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM admin 的密码。
- 对于每个拓扑片段，在 ipatopology_segments 部分添加一个行并设置以下变量：
  - 根据您要添加的分段类型，将 suffix 变量设置为 domain 或 ca。
  - 将 left 变量设置为您要作为复制协议左侧节点的 IdM 服务器的名称。
  - 将 right 变量设置为您要作为复制协议正确节点的 IdM 服务器的名称。

5. 在 check-topologysegments-copy.yml 文件的 tasks 部分中，确保 state 变量设置为 present。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Add topology segments
  hosts: ipaserver
  gather_facts: false

  vars:
    ipaadmin_password: "{{ ipaadmin_password }}"
    ipatopology_segments:
    - {suffix: domain, left: replica1.idm.example.com, right: replica2.idm.example.com }
    - {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
    - {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
    - {suffix: domain+ca, left: replica4.idm.example.com , right:
replica1.idm.example.com }
```

```
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Check topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: "{{ item.suffix }}"
      name: "{{ item.name | default(omit) }}"
      left: "{{ item.left }}"
      right: "{{ item.right }}"
      state: checked
    loop: "{{ ipatopology_segments | default([]) }}"
```

6. 保存该文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory check-
topologysegments-copy.yml
```

**其它资源**

- 有关拓扑协议、后缀和段概念的更多信息，请参阅 解释复制协议、拓扑后缀和拓扑段。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-topology.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/topology 目录中的 playbook 示例。

## 39.4. 使用 ANSIBLE 验证 IDM 中是否存在拓扑后缀

在身份管理 (IdM) 中的复制协议中，拓扑后缀存储要复制的数据。IdM 支持两种类型的拓扑后缀：domain 和 ca。每个后缀代表一个单独的后端，即一个单独的复制拓扑。配置复制协议时，它会在两个不同的服务器上加入同一类型的两个拓扑后缀。

domain 后缀包含与域相关的所有数据，如用户、组和策略。ca 后缀包含证书系统组件的数据。它仅存在于安装有证书颁发机构 (CA) 的服务器上。

按照以下流程，使用 Ansible playbook 确保拓扑后缀在 IdM 中存在。这个示例描述了如何确保 IdM

中存在 domain 后缀。

**先决条件**

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/topology/ 目录中的 verify-topologysuffix.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/ verify-topologysuffix.yml
verify-topologysuffix-copy.yml
```

3. 打开 verify-topologysuffix-copy.yml Ansible playbook 文件进行编辑。

4. 通过在 ipatopologysuffix 部分中设置以下变量来调整文件：

   - 将 ipaadmin_password 变量设置为 IdM admin 的密码。
   - 将 suffix 变量设置为 domain。如果您要验证 ca 后缀是否存在，请将 变量设置为 ca。
   - 确保 state 变量设置为 verify。不允许使用其他选项。

   这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to handle topologysuffix
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Verify topology suffix
    ipatopologysuffix:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      state: verified
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory verify-
topologysuffix-copy.yml
```

**其它资源**

- 请参阅 解释复制协议、拓扑后缀和拓扑段。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-topology.md 文件。

- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/topology 目录中的 playbook 示例。

## 39.5. 使用 ANSIBLE 重新初始化 IDM 副本

如果副本已长时间离线或者其数据库已损坏，您可以重新初始化它。重新初始化会使用更新的一组数据来刷新副本。例如，如果需要从备份进行权威恢复，则可以使用重新初始化。

注意

与复制更新不同，副本仅互相发送更改的条目，重新初始化会刷新整个数据库。

运行命令的本地主机是重新初始化的副本。要指定从中获取数据的副本，请使用 direction 选项。

按照以下流程，使用 Ansible playbook 从 server.idm.example.com 中重新初始化 replica.idm.example.com 上的 domain 数据。

先决条件

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、

服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/topology/ 目录中的 reinitialize-topologysegment.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/reinitialize-
topologysegment.yml reinitialize-topologysegment-copy.yml
```

3. 打开 reinitialize-topologysegment-copy.yml 文件进行编辑。

4. 通过在 ipatopologysegment 部分中设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM admin 的密码。
- 将 suffix 变量设置为 domain。如果您要重新初始化 ca 数据，请将变量设置为 ca。
- 将 left 变量设置为复制协议的左侧节点。
- 将 right 变量设置为复制协议的右节点。
- 将 direction 变量设置为重新初始化数据的方向。left-to-right 方向表示数据从左侧节点流到右侧节点。
- 确保将 state 变量设置为 reinitialized。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to handle topologysegment
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Reinitialize topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      left: server.idm.example.com
      right: replica.idm.example.com
      direction: left-to-right
      state: reinitialized
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory reinitialize-topologysegment-copy.yml
```

**其它资源**

- 请参阅 解释复制协议、拓扑后缀和拓扑段。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-topology.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/topology 目录中的 playbook 示例。

## 39.6. 使用 ANSIBLE 确保 IDM 中没有复制协议

存储在身份管理 (IdM) 服务器上的数据存储基于复制协议：配置了两个服务器时，它们共享其数据。复制协议始终为现实：数据从第一个副本复制到另一个副本，另一个副本复制到第一个副本。

按照以下流程确保两个副本之间的复制协议在 IdM 中不存在。这个示例描述了如何确保在 replica01.idm.example.com 和 replica02.idm.example.com IdM 服务器之间不存在 domain 类型的复制协议。

**先决条件**

- 确保您了解拓扑中连接副本中列出的 IdM 拓扑的建议。
- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/topology/ 目录中的 delete-topologysegment.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/delete-topologysegment.yml
delete-topologysegment-copy.yml
```

3. 打开 delete-topologysegment-copy.yml 文件进行编辑。

4. 通过在 ipatopologysegment 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM admin 的密码。
- 将 suffix 变量设置为 domain。或者，如果您确保 ca 数据不在左侧和右侧节点之间复制，请将变量设置为 ca。
- 将 left 变量设置为您要作为复制协议左侧节点的 IdM 服务器的名称。
- 将 right 变量设置为 IdM 服务器的名称，该服务器是复制协议的右节点。
- 确保 state 变量设置为 absent。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to handle topologysegment
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
- name: Delete topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      left: replica01.idm.example.com
      right: replica02.idm.example.com:
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory delete-
topologysegment-copy.yml
```

**其它资源**

- 请参阅 解释复制协议、拓扑后缀和拓扑段。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-topology.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/topology 目录中的 playbook 示例。

## 39.7. 其它资源

- 请参阅 规划副本拓扑。
- 请参阅 安装 IdM 副本。

# 第 40 章 为用户的外部调配配置 IDM

作为系统管理员，您可以配置身份管理(IdM)，来通过管理身份的外部解决方案支持用户的调配。

外部调配系统的管理员不必使用 ipa 工具，而是使用 ldapmodify 工具来访问 IdM LDAP。管理员可以使用 ldapmodify 从 CLI 或 使用 LDIF 文件 来添加单个 stage 用户。

假设您作为 IdM 管理员完全信任外部调配系统，来仅添加经过验证的用户。但是，您不想为外部调配系统的管理员分配 用户管理员 的 IdM 角色，以便他们能够直接添加新的活动用户。

您可以 配置一个脚本，来自动将外部调配系统创建的 stage 用户移到活动用户。

本章包含以下章节：

1. 准备身份管理(IdM) 来使用外部调配系统向 IdM 添加 stage 用户。
2. 创建一个脚本，来将外部调配系统添加的用户从stage 移到活动用户。
3. 使用外部调配系统添加 IdM stage 用户。您可以通过两种方式进行此操作：
   - 使用 LDIF 文件添加 IdM stage 用户
   - 使用 ldapmodify 直接从 CLI 添加 IdM stage 用户

## 40.1. 为 STAGE 用户帐户的自动激活准备 IDM 帐户

此流程演示了如何配置供外部调配系统使用的两个 IdM 用户帐户。通过使用合适的密码策略将帐户添加到组中，您可以使外部调配系统来管理 IdM 中的用户调配。在以下部分中，外部系统用来添加 stage 用户的用户帐户命名为 provisionator。用来自动激活 stage 用户的用户帐户命名为 activator。

**先决条件**

- 您在其上执行该步骤的主机已注册到 IdM 中。

**流程**

1. 以 IdM 管理员身份登录：

```
$ kinit admin
```

2. 创建名为 provisionator 的用户，其具有用于添加 stage 用户的特权。

a. 添加 provisionator 用户帐户：

```
$ ipa user-add provisionator --first=provisioning --last=account --password
```

a. 为 provisionator 用户授予所需的特权。

i. 创建一个自定义角色 System Provisioning，来管理添加 stage 用户：

```
$ ipa role-add --desc "Responsible for provisioning stage users" "System
Provisioning"
```

ii. 将 Stage User Provisioning 特权添加到该角色。这个特权提供了添加 stage 用户的能力：

```
$ ipa role-add-privilege "System Provisioning" --privileges="Stage User
Provisioning"
```

iii. 将 provisionator 用户添加到角色中：

```
$ ipa role-add-member --users=provisionator "System Provisioning"
```

iv. 验证 provisionator 在 IdM 中是否存在：

```
$ ipa user-find provisionator --all --raw
----------
1 user matched
```

```
--------------
 dn: uid=provisionator,cn=users,cn=accounts,dc=idm,dc=example,dc=com
 uid: provisionator
 [...]
```

3. 创建用户 activator，其具有管理用户帐户的特权。

   a. 添加 activator 用户帐户：

   ```
   $ ipa user-add activator --first=activation --last=account --password
   ```

   b. 通过将用户添加到默认的 User Administrator 角色来授予 activator 用户所需的特权：

   ```
   $ ipa role-add-member --users=activator "User Administrator"
   ```

4. 为应用程序帐户创建用户组：

   ```
   $ ipa group-add application-accounts
   ```

5. 更新组的密码策略。以下策略可防止帐户的密码过期和锁住，但通过要求复杂的密码来弥补潜在的风险：

   ```
   $ ipa pwpolicy-add application-accounts --maxlife=10000 --minlife=0 --history=0 --
   minclasses=4 --minlength=8 --priority=1 --maxfail=0 --failinterval=1 --lockouttime=0
   ```

6. （可选）验证密码策略是否在 IdM 中存在：

   ```
   $ ipa pwpolicy-show application-accounts
     Group: application-accounts
     Max lifetime (days): 10000
     Min lifetime (hours): 0
     History size: 0
   [...]
   ```

7. 将调配和激活帐户添加到应用程序帐户的组中：

   ```
   $ ipa group-add-member application-accounts --users={provisionator,activator}
   ```

8. 更改用户帐户的密码：

```
$ kpasswd provisionator
$ kpasswd activator
```

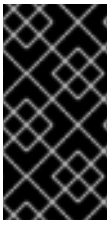
更改密码是必需的，因为新的 IdM 用户密码会立即过期。

其他资源：

- 请参阅 使用命令行管理用户帐户。
- 请参阅 向用户委托权限。
- 请参阅 定义 IdM 密码策略。

### 40.2. 配置 IDM STAGE用户帐户的自动激活

此流程演示了如何为激活 stage 用户创建脚本。系统在指定的时间间隔自动运行脚本。这样可确保新用户帐户被自动激活，并在创建后很快可用。



重要

该流程假定外部调配系统的所有者已经验证了用户，并且在脚本将它们添加到 IdM 之前，它们不需要在 IdM 端进行额外的验证。

这对于仅在一个 IdM 服务器上启用激活过程足够了。

先决条件

- provisionator 和 activator 帐户在 IdM 中存在。详情请参阅 为 stage 用户帐户的自动激活准备 IdM 帐户。
- 在运行该流程的 IdM 服务器上您需要有 root 权限。

- 以 IdM 管理员身份登录。
- 您信任外部调配系统。

**流程**

1. 为激活帐户生成 keytab 文件：

```
# ipa-getkeytab -s server.idm.example.com -p "activator" -k /etc/krb5.ipa-activation.keytab
```

如果您要在多个 IdM 服务器上启用激活过程，请仅在一个服务器上生成 keytab 文件。然后，将 keytab 文件复制到其他服务器上。

2. 创建一个包含以下内容的 /usr/local/sbin/ipa-activate-all 脚本来激活所有用户：

```
#!/bin/bash

kinit -k -i activator

ipa stageuser-find --all --raw | grep "  uid:" | cut -d ":" -f 2 | while read uid; do ipa stageuser-activate ${uid}; done
```

3. 编辑 ipa-activate-all 脚本的权限和所有权来使其可执行：

```
# chmod 755 /usr/local/sbin/ipa-activate-all
# chown root:root /usr/local/sbin/ipa-activate-all
```

4. 创建一个 systemd 单元文件 /etc/systemd/system/ipa-activate-all.service，内容如下：

```
[Unit]
Description=Scan IdM every minute for any stage users that must be activated

[Service]
Environment=KRB5_CLIENT_KTNAME=/etc/krb5.ipa-activation.keytab
Environment=KRB5CCNAME=FILE:/tmp/krb5cc_ipa-activate-all
ExecStart=/usr/local/sbin/ipa-activate-all
```

5. 创建一个 systemd 计时器 /etc/systemd/system/ipa-activate-all.timer，内容如下：

```
[Unit]
Description=Scan IdM every minute for any stage users that must be activated

[Timer]
OnBootSec=15min
OnUnitActiveSec=1min

[Install]
WantedBy=multi-user.target
```

6. 重新载入新配置：

```
# systemctl daemon-reload
```

7. 启用 ipa-activate-all.timer:

```
# systemctl enable ipa-activate-all.timer
```

8. 启动 ipa-activate-all.timer:

```
# systemctl start ipa-activate-all.timer
```

9. （可选）验证 ipa-activate-all.timer 守护进程是否正在运行：

```
# systemctl status ipa-activate-all.timer
● ipa-activate-all.timer - Scan IdM every minute for any stage users that must be
activated
   Loaded: loaded (/etc/systemd/system/ipa-activate-all.timer; enabled; vendor preset:
disabled)
   Active: active (waiting) since Wed 2020-06-10 16:34:55 CEST; 15s ago
  Trigger: Wed 2020-06-10 16:35:55 CEST; 44s left

Jun 10 16:34:55 server.idm.example.com systemd[1]: Started Scan IdM every minute
for any stage users that must be activated.
```

### 40.3. 添加 LDIF 文件中定义的 IDM STAGE 用户

按照以下流程访问 IdM LDAP，并使用 LDIF 文件添加 stage 用户。虽然下例中演示了添加一个单独的用户，但可以以批量模式在一个文件中添加多个用户。

**先决条件**

-

IdM 管理员已为其创建了 provisionator 帐户及密码。详情请参阅 为 stage 用户帐户的自动激活准备 IdM 帐户。

- 作为外部管理员，您知道 provisionator 帐户的密码。
- 您可以从 LDAP 服务器通过 SSH 连接到 IdM 服务器。
- 您可以提供 IdM stage 用户必须有的最小的属性集来允许正确处理用户生命周期，即：
  - 可区分的名称 (dn)
  - 通用名称 (cn)
  - 姓氏 (sn)
  - uid

**流程**

1. 在外部服务器上，创建一个包含有关新用户信息的 LDIF 文件：

```
dn: uid=stageidmuser,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
changetype: add
objectClass: top
objectClass: inetorgperson
uid: stageidmuser
sn: surname
givenName: first_name
cn: full_name
```

2. 将 LDIF 文件从外部服务器传到 IdM 服务器：

```
$ scp add-stageidmuser.ldif provisionator@server.idm.example.com:/provisionator/
Password:
add-stageidmuser.ldif                                         100%  364
217.6KB/s   00:00
```

3. 使用 SSH 协议，以 provisionator 身份连接到 IdM 服务器：

```
$ ssh provisionator@server.idm.example.com
Password:
[provisionator@server ~]$
```

4. 在 IdM 服务器上，获取 provisionator 帐户的 Kerberos 票据授予票(TGT)：

```
[provisionator@server ~]$ kinit provisionator
```

5. 输入 ldapadd 命令，以及 -f 选项和 LDIF 文件的名称。指定 IdM 服务器的名称和端口号：

```
~]$ ldapadd -h server.idm.example.com -p 389 -f  add-stageidmuser.ldif
SASL/GSSAPI authentication started
SASL username: provisionator@IDM.EXAMPLE.COM
SASL SSF: 256
SASL data security layer installed.
adding the entry "uid=stageidmuser,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com"
```

### 40.4. 使用 LDAPMODIFY 直接从 CLI 添加 IDM STAGE 用户

按照以下流程访问身份管理(IdM) LDAP，并使用 ldapmodify 工具添加 stage 用户。

**先决条件**

- IdM 管理员已为其创建了 provisionator 帐户和密码。详情请参阅 为 stage 用户帐户的自动激活准备 IdM 帐户。
- 作为外部管理员，您知道 provisionator 帐户的密码。
- 您可以从 LDAP 服务器通过 SSH 连接到 IdM 服务器。
- 您可以提供 IdM stage 用户必须有的最小的属性集来允许正确处理用户生命周期，即：
  - 可区分的名称 (dn)

- 通用名称 (cn)
- 姓氏 (sn)
- uid

**流程**

1. 使用您的 IdM 身份和凭证，通过 SSH 协议连接到 IdM 服务器：

```
$ ssh provisionator@server.idm.example.com
Password:
[provisionator@server ~]$
```

2. 获取 provisionator 帐户的 TGT，这是具有添加新 stage 用户角色的 IdM 用户：

```
$ kinit provisionator
```

3. 输入 ldapmodify 命令，并将通用安全服务 API(GSSAPI)指定为用于身份验证的简单身份验证和安全层(SASL)机制。指定 IdM 服务器的名称和端口：

```
# ldapmodify -h server.idm.example.com -p 389 -Y GSSAPI
SASL/GSSAPI authentication started
SASL username: provisionator@IDM.EXAMPLE.COM
SASL SSF: 56
SASL data security layer installed.
```

4. 输入您要添加的用户的 dn ：

```
dn: uid=stageuser,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
```

5. 输入 add 作为您要执行的更改的类型：

```
changetype: add
```

6. 指定允许正确处理用户生命周期所需的 LDAP 对象类类别：

```
objectClass: top
objectClass: inetorgperson
```

您可以指定其他对象类。

7. 输入用户的 uid ：

```
uid: stageuser
```

8. 输入用户的 cn:

```
cn: Babs Jensen
```

9. 输入用户的姓氏：

```
sn: Jensen
```

10. 再次按 Enter 键确认输入结束：

```
[Enter]

adding new entry "uid=stageuser,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com"
```

11. 使用 Ctrl + C 退出连接。

验证步骤

验证 stage 条目的内容，以确保您的调配系统添加了所有必需的 POSIX 属性，并且 stage 条目已准备好被激活。

- 要显示新 stage 用户的 LDAP 属性，请输入 ipa stageuser-show --all --raw 命令：

```
$ ipa stageuser-show stageuser --all --raw
 dn: uid=stageuser,cn=staged
```

```
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
 uid: stageuser
 sn: Jensen
 cn: Babs Jensen
 has_password: FALSE
 has_keytab: FALSE
 nsaccountlock: TRUE
 objectClass: top
 objectClass: inetorgperson
 objectClass: organizationalPerson
 objectClass: person
```

1. 请注意，通过 saccountlock 属性，用户被显式禁用了。

## 40.5. 其它资源

- 请参阅 使用 ldapmodify 在外部管理 IdM 用户。

# 第 41 章 使用 LDAPMODIFY 在外部管理 IDM 用户

作为 IdM 管理员，您可以使用 ipa 命令管理您的目录内容。另外，您可以使用 ldapmodify 命令来实现类似的目标。您可以以交互方式使用这个命令，并直接在命令行中提供所有数据。您也可以在 LDAP 数据交换格式(LDIF)的 文件中提供数据到 ldapmodify 命令。

## 41.1. 在外部管理 IDM 用户帐户的模板

以下模板可用于 IdM 中的各种用户管理操作。模板显示您必须使用 ldapmodify 修改哪些属性才能实现以下目标：

- 添加新的 stage 用户
- 修改用户属性
- 启用用户
- 禁用用户
- 保留用户

模板的格式为 LDAP 数据交换格式(LDIF)。LDIF 是一种标准的纯文本数据交换格式，用来表示 LDAP 目录内容和更新请求。

使用模板，您可以配置调配系统的 LDAP 提供者来管理 IdM 用户帐户。

如需详细的示例流程，请参阅以下部分：

- 添加 LDIF 文件中定义的 IdM stage 用户
- 使用 ldapmodify 直接从 CLI 添加 IdM stage 用户

- 使用 ldapmodify 保留 IdM 用户

用于添加新 stage 用户的模板

- 用于添加 自动分配了 UID 和 GID 的用户的模板。所创建的条目的可区别的名称(DN)必须以 uid=user_login 开头：

```
dn: uid=user_login,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
changetype: add
objectClass: top
objectClass: inetorgperson
uid: user_login
sn: surname
givenName: first_name
cn: full_name
```

- 用于添加 静态分配了 UID 和 GID 的用户的模板：

```
dn: uid=user_login,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: inetorgperson
objectClass: organizationalperson
objectClass: posixaccount
uid: user_login
uidNumber: UID_number
gidNumber: GID_number
sn: surname
givenName: first_name
cn: full_name
homeDirectory: /home/user_login
```

在添加 stage 用户时，您不需要指定任何 IdM 对象类。在激活用户后，IdM 自动添加这些类。

用于修改现有用户的模板

- 修改用户的属性 ：

```
dn: distinguished_name
changetype: modify
replace: attribute_to_modify
```

```
attribute_to_modify: new_value
```

- 禁用用户 ：

```
dn: distinguished_name
changetype: modify
replace: nsAccountLock
nsAccountLock: TRUE
```

- 启用用户 ：

```
dn: distinguished_name
changetype: modify
replace: nsAccountLock
nsAccountLock: FALSE
```

更新 nssAccountLock 属性不会对 stage 和preserved 用户造成影响。虽然更新操作成功完成，属性值也会保持 nssAccountLock:TRUE。

- 保留用户 ：

```
dn: distinguished_name
changetype: modrdn
newrdn: uid=user_login
deleteoldrdn: 0
newsuperior: cn=deleted
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
```

注意

在修改用户之前，使用用户的登录名进行搜索来获取用户的可区别名称(DN)。在以下示例中，*user_allowed_to_modify_user_entries* 用户是允许修改用户和组信息的用户，如 activator 或 IdM 管理员。示例中的密码是这个用户的密码：

```
[...]
# ldapsearch -LLL -x -D
"uid=user_allowed_to_modify_user_entries,cn=users,cn=accounts,dc=idm,dc=example,dc=com" -w "Secret123" -H ldap://r8server.idm.example.com -b
"cn=users,cn=accounts,dc=idm,dc=example,dc=com" uid=test_user
dn: uid=test_user,cn=users,cn=accounts,dc=idm,dc=example,dc=com
memberOf: cn=ipausers,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
```

### 41.2. 在外部管理 IDM 组帐户的模板

以下模板可用于 IdM 中的各种用户组管理操作。模板显示您必须使用 ldapmodify 修改哪些属性来实现以下目标：

- 创建新组
- 删除现有组
- 将成员添加到组中
- 从组中删除成员

模板的格式为 LDAP 数据交换格式(LDIF)。LDIF 是一种标准的纯文本数据交换格式，用来表示 LDAP 目录内容和更新请求。

通过使用模板，您可以配置调配系统的 LDAP 提供者来管理 IdM 组帐户。

创建新组

```
dn: cn=group_name,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
changetype: add
objectClass: top
objectClass: ipaobject
objectClass: ipausergroup
objectClass: groupofnames
objectClass: nestedgroup
objectClass: posixgroup
uid: group_name
cn: group_name
gidNumber: GID_number
```

修改组

- 删除现有组 ：

```
dn: group_distinguished_name
changetype: delete
```

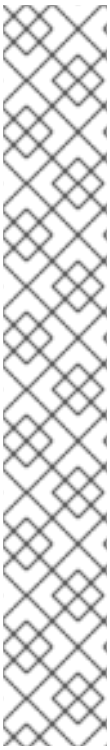
- 将成员添加到组中 ：

```
dn: group_distinguished_name
changetype: modify
add: member
member: uid=user_login,cn=users,cn=accounts,dc=idm,dc=example,dc=com
```

不要向组中添加 stage 或 preserved 的用户。即使更新操作成功完成，也不会作为组的成员更新用户。只有活动的用户才能属于组。

- 从组中删除成员 ：

```
dn: distinguished_name
changetype: modify
delete: member
member: uid=user_login,cn=users,cn=accounts,dc=idm,dc=example,dc=com
```



注意

在修改组之前，使用组的名称进行搜索来获取组的可区别名称(DN)。

```
# ldapsearch -YGSSAPI -H ldap://server.idm.example.com -b
"cn=groups,cn=accounts,dc=idm,dc=example,dc=com" "cn=group_name"
dn: cn=group_name,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
ipaNTSecurityIdentifier: S-1-5-21-1650388524-2605035987-2578146103-11017
cn: testgroup
objectClass: top
objectClass: groupofnames
objectClass: nestedgroup
objectClass: ipausergroup
objectClass: ipaobject
objectClass: posixgroup
objectClass: ipantgroupattrs
ipaUniqueID: 569bf864-9d45-11ea-bea3-525400f6f085
gidNumber: 1997010017
```

### 41.3. 以互动方式使用 LDAPMODIFY 命令

您可以在交互模式中修改轻量级目录访问协议(LDAP)条目。

**流程**

1. 在命令行中，在 ldapmodify 命令后输入 LDAP Data Interchange Format (LDIF)语句。

**例 41.1. 更改 testuser的电话号码**

```
# ldapmodify -Y GSSAPI -H ldap://server.example.com
dn: uid=testuser,cn=users,cn=accounts,dc=example,dc=com
changetype: modify
replace: telephoneNumber
telephonenumber: 88888888
```

请注意，您需要使用 -Y 选项获取 Kerberos ticket。

2. 按 Ctlr+D 退出交互模式。

3. 或者，在 ldapmodify 命令后提供 LDIF 文件：

**例 41.2. ldapmodify 命令从 LDIF 文件中读取修改数据**

```
# ldapmodify -Y GSSAPI -H ldap://server.example.com -f ~/example.ldif
```

**其它资源**

- 有关如何使用 ldapmodify 命令的更多信息，请参阅 ldapmodify (1) 手册页。
- 有关 LDIF 结构的更多信息，请参阅 ldif (5) 手册页。

## 41.4. 使用 LDAPMODIFY 保留 IDM 用户

按照以下流程，使用 ldapmodify 来保留 IdM 用户；即，如何在员工离开公司后停用用户帐户。

**先决条件**

- 您可以作为具有角色的 IdM 用户进行身份验证，来保留用户。

**流程**

1. 以具有角色的 IdM 用户身份登录，来保留用户：

```
$ kinit admin
```

2. 输入 ldapmodify 命令，并指定通用安全服务API(GSSAPI)作为用于身份验证的简单身份验证和安全层(SASL)机制：

```
# ldapmodify -Y GSSAPI
SASL/GSSAPI authentication started
SASL username: admin@IDM.EXAMPLE.COM
SASL SSF: 256
SASL data security layer installed.
```

3. 输入您要保留的用户的 dn：

```
dn: uid=user1,cn=users,cn=accounts,dc=idm,dc=example,dc=com
```

4. 输入 modrdn 作为您要执行的更改的类型：

```
changetype: modrdn
```

5. 为用户指定 newrdn ：

```
newrdn: uid=user1
```

6. 表示您要保留用户：

```
deleteoldrdn: 0
```

7. 指定 新的高级 DN:

```
newsuperior: cn=deleted
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
```

保存用户会将条目移到目录信息树(DIT)中的新位置。因此，您必须将新父条目的 DN 指定为新的高级 DN。

8. 再次按 Enter 键确认输入结束：

```
[Enter]

modifying rdn of entry
"uid=user1,cn=users,cn=accounts,dc=idm,dc=example,dc=com"
```

9. 使用 Ctrl + C 退出连接。

验证步骤

- 通过列出所有 preserved 用户来验证用户是否已保留：

```
$ ipa user-find --preserved=true
--------------
1 user matched
--------------
  User login: user1
  First name: First 1
  Last name: Last 1
  Home directory: /home/user1
  Login shell: /bin/sh
  Principal name: user1@IDM.EXAMPLE.COM
  Principal alias: user1@IDM.EXAMPLE.COM
  Email address: user1@idm.example.com
  UID: 1997010003
  GID: 1997010003
  Account disabled: True
  Preserved user: True
----------------------------
Number of entries returned 1
----------------------------
```

# 第 42 章 在 IDM CLI 中管理主机

本章介绍了身份管理(IdM)中的 主机 和 主机条目，以及在 IdM CLI 中管理主机和主机条目时执行的以下操作：

- 主机注册
- 添加 IdM 主机条目
- 删除 IdM 主机条目
- 重新注册主机
- 重命名主机
- 禁用主机
- 重新启用主机

本章还包含这些操作的前提条件、上下文和结果的 概述表。

## 42.1. IDM 中的主机

Identity Management (IdM) 管理这些身份：

- 用户
- 服务
- 主机

一个主机表示了一个计算机。作为 IdM 身份，主机在 IdM LDAP 中有一个条目，即 IdM 服务器的 389 Directory Server 实例。

IdM LDAP 中的主机条目用于在域中的其他主机甚至服务之间建立关系。这些关系是为域中的主机*委派*授权和控制的一部分。任何主机都可以在基于主机的访问控制 (HBAC) 规则中使用。

IdM 域在计算机之间建立一个通用性，具有通用身份信息、通用策略和共享服务。属于域的任何计算机充当域的客户端，这意味着它使用域所提供的服务。IdM 域为机器提供三个主要服务：

- DNS
- Kerberos
- 证书管理

IdM 中的主机与在其中运行的服务紧密相连：

- 服务条目与主机关联。
- 主机同时存储主机和服务 Kerberos 主体。

## 42.2. 主机注册

本节论述了将主机注册为 IdM 客户端以及注册期间和之后发生的情况。部分比较 IdM 主机和 IdM 用户的注册。部分还概述了可供主机使用的其他身份验证类型。

注册主机包括：

- 在 IdM LDAP 中创建主机条目：可以在 IdM CLI 中使用 ipa host-add 命令，或者等同的 IdM Web UI 操作。
-

在主机上配置 IdM 服务，如系统安全服务守护进程(SSSD)、Kerberos 和 certmonger，并将主机加入 IdM 域。

这两个操作可以单独或一起执行。

如果单独执行，它们允许在具有不同特权级别的两个用户之间划分这两个任务。这对批量部署非常有用。

ipa-client-install 命令可以一起执行两个操作。如果该条目尚不存在，该命令会在 IdM LDAP 中创建主机条目，并为主机配置 Kerberos 和 SSSD 服务。命令将主机引入 IdM 域，并允许它识别它将连接的 IdM 服务器。如果主机属于 IdM 管理的 DNS 区域，ipa-client-install 也为主机添加 DNS 记录。命令必须在客户端上运行。

### 42.3. 主机注册所需的用户权限

主机注册操作需要进行身份验证，以防止非特权用户将不需要的计算机添加到 IdM 域。所需的权限取决于几个因素，例如：

- 创建主机条目与运行 ipa-client-install 是分开的
- 使用一次性密码 (OTP) 进行注册

#### 在 IdM LDAP 中手动创建主机条目的用户权限

使用 ipa host-add CLI 命令或 IdM Web UI 在 IdM LDAP 中创建主机条目所需的用户权限是 Host Administrators。Host Administrators 特权可通过 IT Specialist 角色获得。

#### 将客户端加入 IdM 域的用户特权

在执行 ipa-client-install 命令期间，主机被配置为 IdM 客户端。执行 ipa-client-install 命令所需的凭证级别取决于您发现的以下注册场景：

- IdM LDAP 中的主机条目不存在。在这种情况下，您需要完整的管理员凭据或 Host Administrators 角色。完整的管理员是 admins 组的成员。Host Administrators 角色提供添加主机和注册主机的特权。有关此场景的详情，请参阅 使用用户凭证安装客户端：交互式安装。
- IdM LDAP 中的主机条目存在。在这种情况下，您需要有限的管理员凭证才能成功执行 ipa-

client-install。本例中的有限管理员具有 Enrollment Administrator 角色，该角色提供 Host Enrollment。详情请参阅 使用用户凭证安装客户端：交互式安装。

- IdM LDAP 中的主机条目存在，并且由完整或有限的管理员为主机生成了一个 OTP。在这种情况下，如果您使用 --password 选项运行 ipa-client-install 命令，并提供正确的 OTP，则可以普通用户安装 IdM 客户端。详情请参阅 使用一次性密码安装客户端：交互式安装。

注册后，IdM 主机验证每个新会话，以便能访问 IdM 资源。IdM 服务器需要机器身份验证才能信任机器并接受来自该机器上安装的客户端软件的 IdM 连接。验证客户端后，IdM 服务器可以响应其请求。

## 42.4. IDM 主机和用户的注册和身份验证：比较

IdM 中的用户和主机之间有许多相似之处，其中一些可以在注册阶段观察到，也可以在部署阶段观察到与身份验证有关的相似之处。

- 注册阶段（用户和主机注册）：
  - 管理员可以在用户或主机实际加入 IdM 之前为用户和主机创建 LDAP 条：对于预发布（stage）用户,命令是 ipa stageuser-add ；对于主机，命令是 ipa host-add。
  - 在主机上执行 ipa-client-install 命令时会创建一个包含 *密钥表*（key table，简称为 keytab）和对称密钥（在一定程度上与用户密码相同）的文件，从而使主机可以加入 IdM 域。在逻辑上，用户在激活其帐户时被要求创建密码，因此加入 IdM 域。
  - 虽然用户密码是用户的默认身份验证方法，但 keytab 是主机的默认身份验证方法。keytab 存储在主机上的文件中。

表 42.1. 用户和主机注册

| 操作 | 用户 | 主机 |
| --- | --- | --- |
| 预注册 | $ ipa stageuser-add *user_name* [--password] | $ ipa host-add *host_name* [--random] |
| 激活帐户 | $ ipa stageuser-activate *user_name* | $ ipa-client install [--password] (必需在主机本身上运行) |

- 部署阶段（用户和主机会话身份验证）：
  - 当用户启动新会话时，用户使用密码进行身份验证；类似地，在开机时，主机会通过其 keytab 文件进行身份验证。系统安全服务守护进程 (SSSD) 在后台管理此过程。
  - 如果身份验证成功，用户或主机会获得 Kerberos 票据授予票(TGT)。
  - 然后，使用 TGT 获取特定服务的特定票据。

表 42.2. 用户和主机会话身份验证

| | 用户 | 主机 |
|---|---|---|
| 默认身份验证方式 | 密码 | keytabs |
| 启动会话（普通用户） | $ kinit *user_name* | *[switch on the host]* |
| 身份验证成功的结果 | 用于获取特定服务访问权限的 TGT | 用于获取特定服务访问权限的 TGT |

TGT 和其他 Kerberos 票据作为服务器定义的 Kerberos 服务和策略的一部分生成。IdM 服务会自动授予 Kerberos ticket、更新 Kerberos 凭证甚至销毁 Kerberos 会话。

### IdM 主机的替代身份验证选项

除了 keytabs 外，IdM 还支持两种其他类型的机器验证：

- SSH 密钥。主机的 SSH 公钥已创建并上传到主机条目。从那里，系统安全服务守护进程 (SSSD) 使用 IdM 作为身份提供程序，并可与 OpenSSH 和其他服务一起引用位于 IdM 中的公钥。
- 计算机证书。在这种情况下，计算机使用由 IdM 服务器的证书认证机构签发的 SSL 证书，然后存储在 IdM 的目录服务器中。证书然后发送到计算机，当它向服务器进行身份验证时会存在该证书。在客户端上，证书由名为 certmonger 的服务管理。

## 42.5. 主机操作

**以下部分概述了与主机注册和启用相关的最常见的操作，先决条件、上下文以及执行这些操作的结果。**

**表 42.3. 主机操作第 1 部分**

| 操作 | 操作的先决条件是什么？ | 什么时候运行命令有意义？ | 系统管理员是如何执行操作的？他运行什么命令？ |
|---|---|---|---|
| **注册客户端** | 请参阅 *安装身份管理* 中的 为身份管理客户端安装准备系统 | 当您希望主机加入 IdM 域时。 | 将机器注册为 IdM 域中的客户端是一个两部分的过程。运行 **ipa host-add** 命令时，会为客户端创建一个主机条目（并存储在 389 目录服务器实例中），然后创建一个 keytab 来调配客户端。这两个组件都由 **ipa-client-install** 命令自动执行。也可以单独执行这些步骤；这允许管理员在实际配置客户端之前准备机器和 IdM。这允许更灵活的设置场景，包括批量部署。 |
| **禁用客户端** | 主机必须在 IdM 中有一个条目。主机需要有一个活动的 keytab。 | 可能出于维护目的，您想从 IdM 域临时删除主机。 | **ipa host-disable host_name** |
| **启用客户端** | 主机必须在 IdM 中有一个条目。 | 当您希望临时禁用的主机再次激活时。 | **ipa-getkeytab** |
| **重新注册客户端** | 主机必须在 IdM 中有一个条目。 | 当原始主机丢失，但您已安装了具有相同主机名的主机时。 | **ipa-client-install --keytab** 或 **ipa-client-install --force-join** |
| **取消注册客户端** | 主机必须在 IdM 中有一个条目。 | 当您要从 IdM 域永久删除主机时： | **ipa-client-install --uninstall** |

**表 42.4. 主机操作第 2 部分**

| 操作 | 管理员可以在哪一台机器上运行命令？ | 执行该操作时会发生什么情况？主机在 IdM 中正常工作的结果是什么？引入了/删除了哪些限制？ |
|---|---|---|
| **注册客户端** | 对于两步注册：**ipa host-add** 可以运行在任何一台 IdM 客户端上；**ipa-client-install** 的第二步必须运行在客户端本身上 | 默认情况下，这会将 SSSD 配置为连接到 IdM 服务器来进行身份验证和授权。另外，也可以将可插拔验证模块(PAM)和名称交换服务(NSS)配置为通过 Kerberos 和 LDAP 与 IdM 服务器一起工作。 |

| 操作 | 管理员可以在哪一台机器上运行命令？ | 执行该操作时会发生什么情况？主机在 IdM 中正常工作的结果是什么？引入了/删除了哪些限制？ |
| --- | --- | --- |
| **禁用客户端** | IdM 中的任何机器，即使主机本身 | 主机的 Kerberos 密钥和 SSL 证书无效，运行在该主机上的所有服务都被禁用。 |
| **启用客户端** | IdM 中的任何机器。如果在禁用的主机上运行，则需要提供 LDAP 凭据。 | 主机的 Kerberos 密钥和 SSL 证书将再次有效，所有运行在主机上的 IdM 服务都被重新启用。 |
| **重新注册客户端** | 重新注册的主机。需要提供 LDAP 凭据。 | 为主机生成一个新的 Kerberos 密钥，替换之前的密钥。 |
| **取消注册客户端** | 要取消注册的主机。 | 命令取消配置 IdM，并尝试将机器返回到之前的状态。此过程的一部分是从 IdM 服务器取消注册主机。取消注册包括在 IdM 服务器上禁用主密钥。**/etc/krb5.keytab**(**host/<fqdn>@REALM**) 中的机器主体用于向 IdM 服务器进行身份验证以取消注册。如果这个主体不存在，则取消注册会失败，管理员将需要禁用主机主体(**ipa host-disable <fqdn>**)。 |

### 42.6. IDM LDAP 中的主机条目

**身份管理(IdM)主机条目包含有关主机的信息及其可以包含哪些属性。**

**LDAP 主机条目包含 IdM 中关于客户端的所有相关信息：**

- **与主机关联的服务条目**
- **主机和服务主体**
- **访问控制规则**
- **机器信息，如物理位置和操作系统**

**注意**

请注意，IdM Web UI Identity → Hosts 选项卡不会显示有关存储在 IdM LDAP 中的特定主机的所有信息。

### 主机条目配置属性

主机条目可以包含其系统配置之外的主机的信息，如其物理位置、MAC 地址、密钥和证书。

如果主机条目是手动创建的，则可在创建主机条目时设置此信息。另外，大多数此类信息可以在主机注册到域后添加到主机条目中。

**表 42.5. 主机配置属性**

| UI 字段 | 命令行选项 | 描述 |
|---|---|---|
| 描述 | **--desc**=*description* | 主机的描述。 |
| 地点 | **--locality**=*locality* | 主机的地理位置。 |
| 位置 | **--location**=*location* | 主机的物理位置，如其数据中心机架。 |
| 平台 | **--platform**=*string* | 主机硬件或架构. |
| 操作系统 | **--os**=*string* | 主机的操作系统和版本。 |
| MAC 地址 | **--macaddress**=*address* | 主机的 MAC 地址。这是一个多值属性。NIS 插件使用 MAC 地址为主机创建 NIS **ethers** 映射。 |
| SSH 公钥 | **--sshpubkey**=*string* | 主机的完整 SSH 公钥。这是一个多值属性，因此可以设置多个键。 |
| 主体名称（不可编辑） | **--principalname**=*principal* | 主机的 Kerberos 主体名称。除非在 **-p** 中显式设置了不同的主体，否则默认为客户端安装期间的主机名。这可以通过命令行工具进行更改，但不能在 UI 中更改。 |
| 设置一次性密码 | **--password**=*string* | 此选项为可用于批量注册的主机设置密码。 |
| - | **--random** | 此选项生成一个用于批量注册的随机密码。 |

| UI 字段 | 命令行选项 | 描述 |
| --- | --- | --- |
| - | **--certificate**=*string* | 主机的证书 blob。 |
| - | **--updatedns** | 这会设置主机在其 IP 地址更改时是否可以动态更新其 DNS 条目。 |

### 42.7. 从 IDM CLI 添加 IDM 主机条目

按照以下流程，使用命令行界面(CLI)在身份管理(IdM)中添加主机条目。

主机条目使用 **host-add** 命令来创建。此命令将主机条目添加到 IdM 目录服务器中。通过在 CLI 中输入 **ipa help host** 来查阅 **ipa host** 手册页，以获取 **host-add** 可用选项的完整列表。

向 IdM 添加主机时有几个不同的场景：

- 最基本的场景，仅指定客户端主机名来将客户端添加到 Kerberos 域，并在 IdM LDAP 服务器中创建一个条目：

```
$ ipa host-add client1.example.com
```

- 如果 IdM 服务器被配置为管理 DNS，请使用 **--ip-address** 选项将主机添加到 DNS 资源记录中。

**例 42.1. 创建具有静态 IP 地址的主机条目**

```
$ ipa host-add --ip-address=192.168.166.31 client1.example.com
```

- 如果要添加的主机没有静态 IP 地址，或者在配置客户端时不知道 IP 地址，请使用 **ipa host-add** 命令的 **--force** 选项。

**例 42.2. 创建具有 DHCP 的主机条目**

```
$ ipa host-add --force client1.example.com
```

例如，笔记本电脑可能预配置为 IdM 客户端，但它们在配置时没有 IP 地址。使用 **--force** 实际上是在 IdM DNS 服务中创建一个占位符条目。当 DNS 服务动态更新其记录时，将检测主机的

当前 IP 地址，并更新其 DNS 记录。

## 42.8. 从 IDM CLI 删除主机条目

- 使用 host-del 命令删除主机记录。如果您的 IdM 域已集成了 DNS，请使用 --updatedns 选项从 DNS 中删除主机任何类型的关联记录：

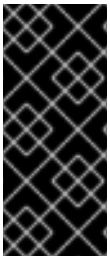
```
$ ipa host-del --updatedns client1.example.com
```

## 42.9. 重新注册身份管理客户端

本节描述了重新注册身份管理客户端的不同方法。

### 42.9.1. IdM 中的客户端重新注册

在重新注册过程中，客户端会生成一个新的 Kerberos 密钥和 SSH 密钥，但 LDAP 数据库中客户端的身份保持不变。重新注册后，在机器与 IdM 服务器失去连接之前，主机像以前一样，其密钥和其他信息放在具有相同 FQDN 的同一 LDAP 对象中。



重要

您只能重新注册域条目仍然活跃的客户端。如果您卸载了客户端（使用 ipa-client-install --uninstall）或者禁用了其主机条目（使用 ipa host-disable），则无法重新注册它。

您不能在重命名客户端后重新注册客户端。这是因为在身份管理中，LDAP 中客户端条目的 key 属性是客户端的主机名，即其 FQDN。与重新注册客户端（在此期间客户端的 LDAP 对象保持不变）不同，重命名客户端的结果是，客户端的密钥和其他信息位于具有新 FQDN 的不同的 LDAP 对象中。因此，重命名客户端的唯一方法是从 IdM 卸载主机，更改主机的主机名，并使用新名称将其安装为 IdM 客户端。有关如何重命名客户端的详情，请参考 重命名身份管理客户端系统。

客户端重新注册过程中会发生什么

重新注册期间的身份管理：

- 吊销原始主机证书

- 创建新 SSH 密钥
- 生成一个新的 keytab

### 42.9.2. 使用用户凭证重新注册客户端： 交互式重新注册

按照以下流程，使用授权用户的凭证以互动方式重新注册身份管理客户端。

1. 重新创建具有相同主机名的客户端机器。
2. 在客户端机器上运行 ipa-client-install --force-join 命令：

```
# ipa-client-install --force-join
```

3. 该脚本提示其身份用于重新注册客户端的用户。例如，这可能是具有注册管理员角色的 hostadmin 用户：

```
User authorized to enroll computers: hostadmin
Password for hostadmin@EXAMPLE.COM:
```

**其它资源**

- 请参阅 *安装身份管理* 中的 使用用户凭证安装客户端：交互式安装。

### 42.9.3. 使用 client keytab: Non-interactive reenrollment 重新注册客户端

**先决条件**

- 备份原始客户端 keytab 文件，例如在 /tmp 或 /root 目录中。

**流程**

按照以下流程，使用客户端系统的 keytab 以非交互方式重新注册身份管理(IdM)客户端。例如，使用客户端 keytab 重新注册适用于自动安装。

1. 重新创建具有相同主机名的客户端机器。

2. 将 keytab 文件从备份位置复制到重新创建的客户端机器上的 /etc/ 目录。

3. 使用 ipa-client-install 工具重新注册客户端，并使用 --keytab 选项指定 keytab 的位置：

```
# ipa-client-install --keytab /etc/krb5.keytab
```

**注意**

--keytab 选项中指定的 keytab只在进行身份验证以启动注册时才使用。在重新注册过程中，IdM 为客户端生成一个新的 keytab。

### 42.9.4. 安装后测试身份管理客户端

命令行界面告知您 ipa-client-install 已成功，但您也可以自行进行测试。

要测试身份管理客户端是否可以获取服务器上定义的用户的信息，请检查您是否能够解析服务器上定义的用户。例如，检查默认的 admin 用户：

```
[user@client1 ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

要测试身份验证是否正常工作，请su - 为另一个 IdM 用户：

```
[user@client1 ~]$ su - idm_user
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[idm_user@client1 ~]$
```

## 42.10. 重命名身份管理客户端系统

以下章节描述了如何更改身份管理客户端系统的主机名。

**警告**

重新命名客户端是一个手动过程。除非绝对需要修改主机名，否则请勿执行此操作。

**重命名身份管理客户端涉及：**

1. 准备主机。详情请参阅 准备 IdM 客户端以进行重命名。
2. 从主机卸载 IdM 客户端。详情请参阅 卸载身份管理客户端。
3. 重命名主机。详情请参阅 重命名主机系统。
4. 使用新名称在主机上安装 IdM 客户端。详情请参阅 *安装身份管理* 中的安装身份管理客户端。
5. 在 IdM 客户端安装后配置主机。详情请查看 重新添加服务、重新生成证书和重新添加主机组。

### 42.10.1. 准备 IdM 客户端以进行重命名

在卸载当前客户端之前，请记下客户端的某些设置。在使用新的主机名重新注册计算机后，您将应用此配置。

- 确定在机器上运行哪些服务：
  - 使用 ipa service-find 命令，并在输出中识别带有证书的服务：

```
$ ipa service-find old-client-name.example.com
```

此外，每个主机都有一个默认 *主机服务*，该服务不会出现在 ipa service-find 输出中。主机服务的服务主体（也称为 *主机主体*）是 host/*old-client-name.example.com*。

- 对于 ipa service-find *old-client-name.example.com* 显示的所有服务主体，请确定 *old-client-name.example.com* 系统上相应的 keytab 的位置：

```
# find / -name "*.keytab"
```

客户端系统上的每个服务都有一个格式为 *service_name/host_name@REALM* 的 Kerberos 主体，例如 ldap/*old-client-name.example.com@EXAMPLE.COM*.

- 识别机器所属的所有主机组。

```
# ipa hostgroup-find old-client-name.example.com
```

### 42.10.2. 卸载身份管理客户端

卸载客户端会从身份管理域中删除客户端，以及系统服务的所有特定身份管理配置，如系统安全服务守护进程(SSSD)。这会恢复客户端系统的以前的配置。

**流程**

1. 运行 ipa-client-install --uninstall 命令：

```
[root@client]# ipa-client-install --uninstall
```

2. 从服务器中手动删除客户端主机的 DNS 条目：

```
[root@server]# ipa dnsrecord-del
Record name: old-client-client
Zone name: idm.example.com
No option to delete specific record provided.
Delete all? Yes/No (default No): yes
----------------------
Deleted record "old-client-name"
```

3. 对于除 /etc/krb5.keytab 以外的每个识别的 keytab，删除旧的主体：

```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

4. 在 IdM 服务器上，删除主机条目。这会删除所有服务并吊销为该主机发布的所有证书：

```
[root@server ~]# ipa host-del client.example.com
```

### 42.10.3. 重命名主机系统

根据需要重命名机器。例如：

```
[root@client]# hostnamectl set-hostname new-client-name.example.com
```

现在，您可以使用新的主机名将身份管理客户端重新安装到身份管理域中。

### 42.10.4. 重新添加服务、重新生成证书和重新添加主机组

**流程**

1. 在身份管理(IdM)服务器上，为 准备 IdM 客户端以进行重命名 中指定的每个服务添加新的 keytab 。

```
[root@server ~]# ipa service-add service_name/new-client-name
```

2. 为 准备 IdM 客户端以进行重命名 中分配了证书的服务生成证书。您可以做到这一点：

   - 使用 IdM 管理工具
   - 使用 certmonger 工具

3. 将客户端重新添加到 准备 IdM 客户端以进行重命名 中标识的主机组。

## 42.11. 禁用和重新启用主机条目

本节介绍了如何在身份管理(IdM)中禁用和重新启用主机。

### 42.11.1. 禁用主机

完成这个流程来禁用 IdM 中的主机条目。

域服务、主机和用户可以访问活动的主机。某些情况下，出于维护原因需要临时删除活动的主机。在这种情况下，不需要删除主机，因为它会永久删除主机条目和所有关联的配置。相反，可选择禁用该主机的选项。

禁用主机可防止域用户访问它，而不必将其从域中永久删除。

流程

- 使用 host-disable 命令禁用主机。禁用主机将终止主机当前活动的 keytab。例如：

```
$ kinit admin
$ ipa host-disable client.example.com
```

禁用主机后，主机将对所有 IdM 用户、主机和服务都不可用。

重要

禁用主机条目不仅会禁用该主机。它还会禁用该主机上每个配置的服务。

### 42.11.2. 重新启用主机

按照以下流程重新启用禁用的 IdM 主机。

禁用主机会终止其活动的 keytab，这会从 IdM 域中删除主机，而不影响其配置条目。

流程

- 要重新启用主机，请使用 ipa-getkeytab 命令，添加：
  - -s 选项来指定要从哪个 IdM 服务器请求 keytab

- -p 选项来指定主体名称

- k 选项来指定保存 keytab 的文件。

例如，要为 client.example.com 从 server.example.com 请求新的主机 keytab，并将 keytab 存储在 /etc/krb5.keytab 文件中：

```
$ ipa-getkeytab -s server.example.com -p host/client.example.com -k /etc/krb5.keytab -D "cn=directory manager" -w password
```

注意

您还可以使用管理员的凭据，指定 -D "uid=admin,cn=users,cn=accounts,dc=example,dc=com"。重要的是，凭据对应于允许为主机创建 keytab 的用户。

如果 ipa-getkeytab 命令在活动的 IdM 客户端或服务器上运行，那么如果用户具有例如通过kinit admin 获取的 TGT，则可以在没有 LDAP 凭据（-D 和 -w ）的情况下运行该命令。若要在禁用的主机上直接运行命令，请提供 LDAP 凭据来向 IdM 服务器进行身份验证。

# 第 43 章 从 IDM WEB UI 添加主机条目

本章介绍了身份管理(IdM)中的主机，以及在 IdM Web UI 中添加主机条目的操作。

## 43.1. IDM 中的主机

Identity Management (IdM) 管理这些身份：

- 用户
- 服务
- 主机

一个主机表示了一个计算机。作为 IdM 身份，主机在 IdM LDAP 中有一个条目，即 IdM 服务器的 389 Directory Server 实例。

IdM LDAP 中的主机条目用于在域中的其他主机甚至服务之间建立关系。这些关系是为域中的主机*委派*授权和控制的一部分。任何主机都可以在基于主机的访问控制 (HBAC) 规则中使用。

IdM 域在计算机之间建立一个通用性，具有通用身份信息、通用策略和共享服务。属于域的任何计算机充当域的客户端，这意味着它使用域所提供的服务。IdM 域为机器提供三个主要服务：

- DNS
- Kerberos
- 证书管理

IdM 中的主机与在其中运行的服务紧密相连：

- 服务条目与主机关联。
- 主机同时存储主机和服务 Kerberos 主体。

## 43.2. 主机注册

本节论述了将主机注册为 IdM 客户端以及注册期间和之后发生的情况。部分比较 IdM 主机和 IdM 用户的注册。部分还概述了可供主机使用的其他身份验证类型。

注册主机包括：

- 在 IdM LDAP 中创建主机条目：可以在 IdM CLI 中使用 ipa host-add 命令，或者等同的 IdM Web UI 操作。
- 在主机上配置 IdM 服务，如系统安全服务守护进程(SSSD)、Kerberos 和 certmonger，并将主机加入 IdM 域。

这两个操作可以单独或一起执行。

如果单独执行，它们允许在具有不同特权级别的两个用户之间划分这两个任务。这对批量部署非常有用。

ipa-client-install 命令可以一起执行两个操作。如果该条目尚不存在，该命令会在 IdM LDAP 中创建主机条目，并为主机配置 Kerberos 和 SSSD 服务。命令将主机引入 IdM 域，并允许它识别它将连接的 IdM 服务器。如果主机属于 IdM 管理的 DNS 区域，ipa-client-install 也为主机添加 DNS 记录。命令必须在客户端上运行。

## 43.3. 主机注册所需的用户权限

主机注册操作需要进行身份验证，以防止非特权用户将不需要的计算机添加到 IdM 域。所需的权限取决于几个因素，例如：

- 创建主机条目与运行 ipa-client-install 是分开的

- 使用一次性密码 (OTP) 进行注册

在 IdM LDAP 中手动创建主机条目的用户权限

使用 ipa host-add CLI 命令或 IdM Web UI 在 IdM LDAP 中创建主机条目所需的用户权限是 Host Administrators。Host Administrators 特权可通过 IT Specialist 角色获得。

将客户端加入 IdM 域的用户特权

在执行 ipa-client-install 命令期间，主机被配置为 IdM 客户端。执行 ipa-client-install 命令所需的凭证级别取决于您发现的以下注册场景：

- IdM LDAP 中的主机条目不存在。在这种情况下，您需要完整的管理员凭据或 Host Administrators 角色。完整的管理员是 admins 组的成员。Host Administrators 角色提供添加主机和注册主机的特权。有关此场景的详情，请参阅 使用用户凭证安装客户端：交互式安装。
- IdM LDAP 中的主机条目存在。在这种情况下，您需要有限的管理员凭证才能成功执行 ipa-client-install。本例中的有限管理员具有 Enrollment Administrator 角色，该角色提供 Host Enrollment。详情请参阅 使用用户凭证安装客户端：交互式安装。
- IdM LDAP 中的主机条目存在，并且由完整或有限的管理员为主机生成了一个 OTP。在这种情况下，如果您使用 --password 选项运行 ipa-client-install 命令，并提供正确的 OTP，则可以普通用户安装 IdM 客户端。详情请参阅 使用一次性密码安装客户端：交互式安装。

注册后，IdM 主机验证每个新会话，以便能访问 IdM 资源。IdM 服务器需要机器身份验证才能信任机器并接受来自该机器上安装的客户端软件的 IdM 连接。验证客户端后，IdM 服务器可以响应其请求。

## 43.4. IDM 主机和用户的注册和身份验证：比较

IdM 中的用户和主机之间有许多相似之处，其中一些可以在注册阶段观察到，也可以在部署阶段观察到与身份验证有关的相似之处。

- 注册阶段（用户和主机注册）：
  - 管理员可以在用户或主机实际加入 IdM 之前为用户和主机创建 LDAP 条：对于预发布（stage）用户,命令是 ipa stageuser-add ；对于主机，命令是 ipa host-add。

- 在主机上执行 ipa-client-install 命令时会创建一个包含 *密钥表*（key table，简称为 keytab）和对称密钥（在一定程度上与用户密码相同）的文件，从而使主机可以加入 IdM 域。在逻辑上，用户在激活其帐户时被要求创建密码，因此加入 IdM 域。

- 虽然用户密码是用户的默认身份验证方法，但 keytab 是主机的默认身份验证方法。keytab 存储在主机上的文件中。

**表 43.1. 用户和主机注册**

| 操作 | 用户 | 主机 |
|---|---|---|
| 预注册 | $ ipa stageuser-add *user_name* [--password] | $ ipa host-add *host_name* [--random] |
| 激活帐户 | $ ipa stageuser-activate *user_name* | $ ipa-client install [--password] (必需在主机本身上运行) |

- 部署阶段（用户和主机会话身份验证）：

  - 当用户启动新会话时，用户使用密码进行身份验证；类似地，在开机时，主机会通过其 keytab 文件进行身份验证。系统安全服务守护进程 (SSSD) 在后台管理此过程。

  - 如果身份验证成功，用户或主机会获得 Kerberos 票据授予票(TGT)。

  - 然后，使用 TGT 获取特定服务的特定票据。

**表 43.2. 用户和主机会话身份验证**

| | 用户 | 主机 |
|---|---|---|
| 默认身份验证方式 | **密码** | **keytabs** |
| 启动会话（普通用户） | $ **kinit** ***user_name*** | *[switch on the host]* |
| 身份验证成功的结果 | 用于获取特定服务访问权限的 TGT | 用于获取特定服务访问权限的 TGT |

TGT 和其他 Kerberos 票据作为服务器定义的 Kerberos 服务和策略的一部分生成。IdM 服务会自动授予 Kerberos ticket、更新 Kerberos 凭证甚至销毁 Kerberos 会话。

### IdM 主机的替代身份验证选项

除了 keytabs 外，IdM 还支持两种其他类型的机器验证：

- SSH 密钥。主机的 SSH 公钥已创建并上传到主机条目。从那里，系统安全服务守护进程 (SSSD) 使用 IdM 作为身份提供程序，并可与 OpenSSH 和其他服务一起引用位于 IdM 中的公钥。
- 计算机证书。在这种情况下，计算机使用由 IdM 服务器的证书认证机构签发的 SSL 证书，然后存储在 IdM 的目录服务器中。证书然后发送到计算机，当它向服务器进行身份验证时会存在该证书。在客户端上，证书由名为 certmonger 的服务管理。

## 43.5. IDM LDAP 中的主机条目

身份管理(IdM)主机条目包含有关主机的信息及其可以包含哪些属性。

LDAP 主机条目包含 IdM 中关于客户端的所有相关信息：

- 与主机关联的服务条目
- 主机和服务主体
- 访问控制规则
- 机器信息，如物理位置和操作系统

**注意**

请注意，IdM Web UI Identity → Hosts 选项卡不会显示有关存储在 IdM LDAP 中的特定主机的所有信息。

### 主机条目配置属性

主机条目可以包含其系统配置之外的主机的信息，如其物理位置、MAC 地址、密钥和证书。

**如果主机条目是手动创建的，则可在创建主机条目时设置此信息。另外，大多数此类信息可以在主机注册到域后添加到主机条目中。**

**表 43.3. 主机配置属性**

| UI 字段 | 命令行选项 | 描述 |
| --- | --- | --- |
| 描述 | **--desc**=*description* | 主机的描述。 |
| 地点 | **--locality**=*locality* | 主机的地理位置。 |
| 位置 | **--location**=*location* | 主机的物理位置，如其数据中心机架。 |
| 平台 | **--platform**=*string* | 主机硬件或架构. |
| 操作系统 | **--os**=*string* | 主机的操作系统和版本。 |
| MAC 地址 | **--macaddress**=*address* | 主机的 MAC 地址。这是一个多值属性。NIS 插件使用 MAC 地址为主机创建 NIS **ethers** 映射。 |
| SSH 公钥 | **--sshpubkey**=*string* | 主机的完整 SSH 公钥。这是一个多值属性，因此可以设置多个键。 |
| 主体名称（不可编辑） | **--principalname**=*principal* | 主机的 Kerberos 主体名称。除非在 **-p** 中显式设置了不同的主体，否则默认为客户端安装期间的主机名。这可以通过命令行工具进行更改，但不能在 UI 中更改。 |
| 设置一次性密码 | **--password**=*string* | 此选项为可用于批量注册的主机设置密码。 |
| - | **--random** | 此选项生成一个用于批量注册的随机密码。 |
| - | **--certificate**=*string* | 主机的证书 blob。 |
| - | **--updatedns** | 这会设置主机在其 IP 地址更改时是否可以动态更新其 DNS 条目。 |

### 43.6. 从 WEB UI 添加主机条目

1. **打开 Identity 选项卡，然后选择 Hosts 子选项卡。**

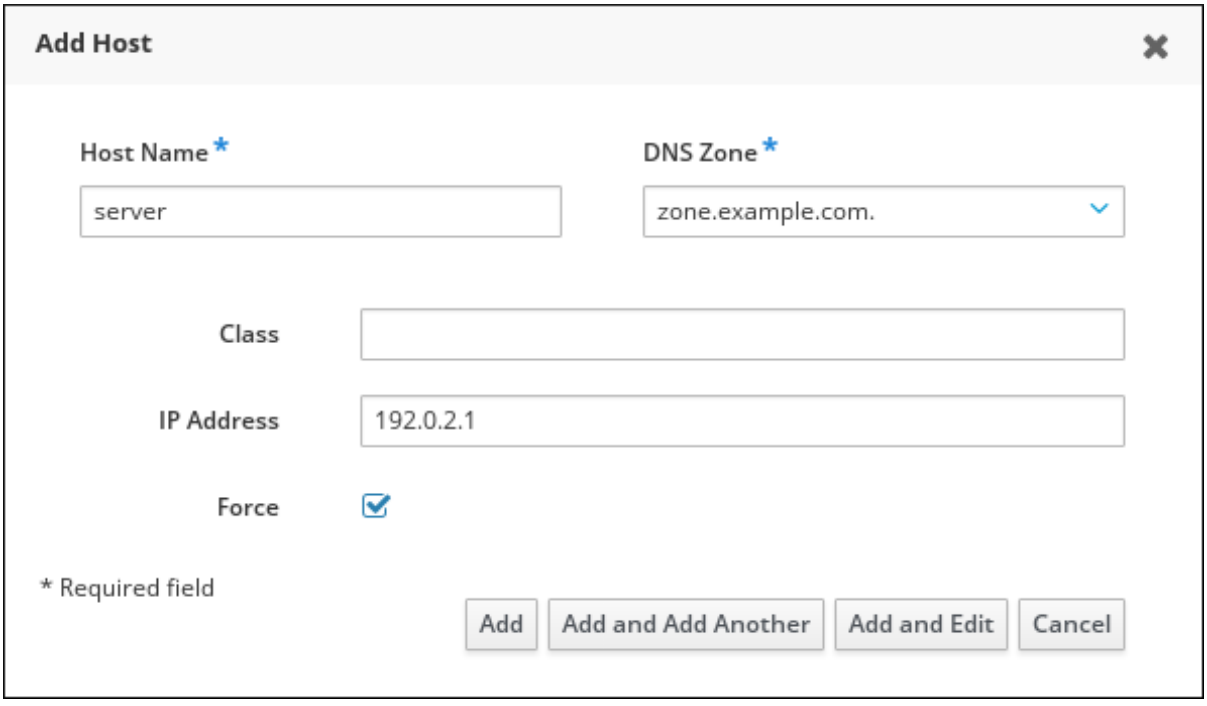
2. 单击主机列表顶部的 Add。

图 43.1. 添加主机条目


3. 输入机器名称，并在下拉列表中配置的区中选择域。如果已经为主机分配了静态 IP 地址，则将它与主机条目一起包含，以便完全创建 DNS 条目。

Class 字段目前没有特定的目的。

图 43.2. 添加主机向导



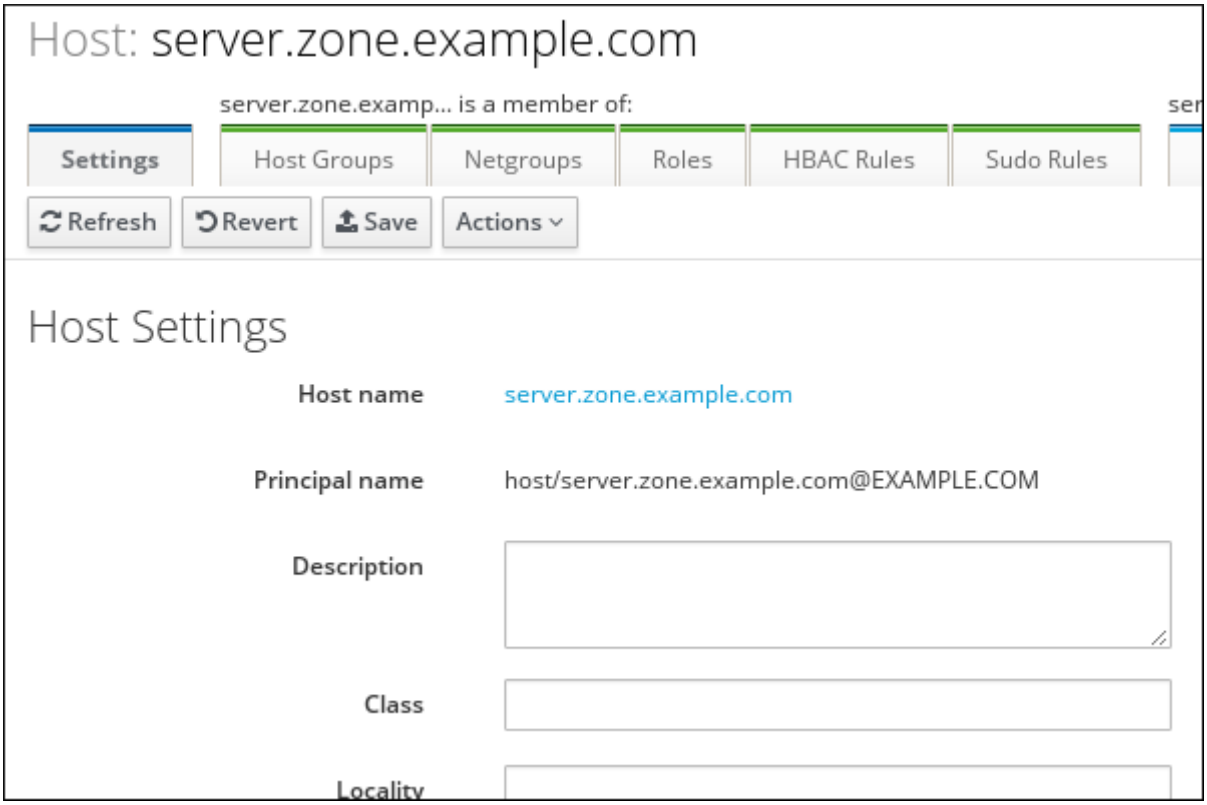

可以在 IdM 中创建 DNS 区。如果 IdM 服务器不管理 DNS 服务器，则可以在菜单区域中手动输入区，如常规文本字段。

**注意**

如果要跳过检查主机是否可以通过 DNS 解析，请选择 Force 复选框。

4. 单击 **Add and Edit** 按钮，直接进入到扩展的条目页面，输入更多的属性信息。有关主机硬件和物理位置的信息可以包含在主机条目中。

**图 43.3. 扩展的条目页面**



# 第 44 章 使用 ANSIBLE PLAYBOOK 管理主机

Ansible 是一个自动化工具，用于配置系统、部署软件和执行滚动更新。Ansible 包含对身份管理 (IdM) 的支持，您可以使用 Ansible 模块自动执行主机管理。

在使用 Ansible playbook 管理主机和主机条目时，将执行以下概念和操作：

- 确保存在的 IdM 主机条目仅由 FQDN定义
- 确保存在带有 IP 地址的 IdM 主机条目
- 确保存在带有随机密码的多个 IdM 主机条目
- 确保存在带有多个 IP 地址的 IdM 主机条目
- 确保 IdM 主机条目不存在

## 44.1. 使用 ANSIBLE PLAYBOOK 确保存在带有 FQDN 的 IDM 主机条目

按照以下流程，使用 Ansible playbook 确保主机条目在身份管理(IdM)中存在。主机条目仅通过其 完全限定域名 (FQDN) 定义。

如果至少适用以下条件之一，则指定主机的 FQDN 名称就足够：

- IdM 服务器没有配置为管理 DNS。
- 主机没有静态 IP 地址，或者在配置主机时不知道该 IP 地址。添加仅由 FQDN 定义的主机实质上会在 IdM DNS 服务中创建占位符条目。例如，笔记本电脑可能预配置为 IdM 客户端，但它们在配置时没有 IP 地址。当 DNS 服务动态更新其记录时，将检测主机的当前 IP 地址，并更新其 DNS 记录。

**注意**

如果没有 Ansible，则使用 ipa host-add 命令在 IdM 中创建主机条目。将主机添加到 IdM 的结果是 IdM 中存在的主机状态。由于 Ansible 依赖于 idempotence，要使用 Ansible 将主机添加到 IdM，您必须创建一个 playbook，将主机的状态定义为 present: state: present。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

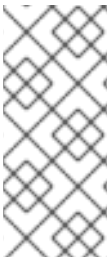
```
[ipaserver]
server.idm.example.com
```

2. 创建一个 Ansible playbook 文件，其中包含您要确保的 IdM 中的 FQDN。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/host/add-host.yml 文件中的示例：

```
---
- name: Host present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Host host01.idm.example.com present
    ipahost:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: host01.idm.example.com
      state: present
      force: yes
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-host-
is-present.yml
```



**注意**

这个过程会导致在 IdM LDAP 服务器中创建主机条目，但不将主机注册到 IdM Kerberos 域。为此，您必须将主机部署为 IdM 客户端。详情请参阅使用 Ansible playbook 安装身份管理客户端。

**验证步骤**

1. 以 admin 用户身份登录您的 IdM 服务器：

```
$ ssh admin@server.idm.example.com
Password:
```

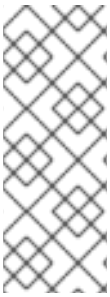
2. 输入 ipa host-show 命令并指定主机名称：

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Password: False
  Keytab: False
  Managed by: host01.idm.example.com
```

输出确认 IdM 中存在 host01.idm.example.com。

## 44.2. 使用 ANSIBLE PLAYBOOK 确保存在含有 DNS 信息的 IDM 主机条目

按照以下流程，使用 Ansible playbook 确保主机条目在身份管理(IdM)中存在。主机条目通过其 完全限定域名 (FQDN)及其 IP 地址定义。



**注意**

如果没有 Ansible，则使用 ipa host-add 命令在 IdM 中创建主机条目。将主机添加到 IdM 的结果是 IdM 中存在的主机状态。由于 Ansible 依赖于 idempotence，要使用 Ansible 将主机添加到 IdM，您必须创建一个 playbook，将主机的状态定义为 present: state: present。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 创建一个 Ansible playbook 文件，其中包含您要确保的 IdM 中的 完全限定域名 (FQDN)。另外，如果 IdM 服务器配置为管理 DNS，并且您知道主机的 IP 地址，请为 ip_address 参数指定一个值。主机需要 IP 地址才能存在于 DNS 资源记录中。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/host/host-present.yml 文件中的示例。您还可以包含其他附加信息：

```
---
- name: Host present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure host01.idm.example.com is present
    ipahost:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: host01.idm.example.com
      description: Example host
      ip_address: 192.168.0.123
      locality: Lab
      ns_host_location: Lab
      ns_os_version: CentOS 7
      ns_hardware_platform: Lenovo T61
      mac_address:
      - "08:00:27:E3:B1:2D"
      - "52:54:00:BD:97:1E"
      state: present
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-host-
is-present.yml
```

**注意**

这个过程会导致在 IdM LDAP 服务器中创建主机条目，但不将主机注册到 IdM Kerberos 域。为此，您必须将主机部署为 IdM 客户端。详情请参阅使用 Ansible playbook 安装身份管理客户端。

验证步骤

1. 以 admin 用户身份登录您的 IdM 服务器：

```
$ ssh admin@server.idm.example.com
Password:
```

2. 输入 ipa host-show 命令并指定主机名称：

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Description: Example host
  Locality: Lab
  Location: Lab
  Platform: Lenovo T61
  Operating system: CentOS 7
  Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
  MAC address: 08:00:27:E3:B1:2D, 52:54:00:BD:97:1E
  Password: False
  Keytab: False
  Managed by: host01.idm.example.com
```

输出确认 IdM 中存在 host01.idm.example.com。

## 44.3. 使用 ANSIBLE PLAYBOOK 确保存在带有随机密码的多个 IDM 主机条目

ipahost 模块允许系统管理员使用一个 Ansible 任务来确保 IdM 中存在或不存在多个主机条目。按照以下流程，确保仅由 完全限定域名 (FQDN)定义的多个主机条目存在。运行 Ansible playbook 会为主机生成随机密码。

**注意**

如果没有 Ansible，则使用 ipa host-add 命令在 IdM 中创建主机条目。将主机添加到 IdM 的结果是 IdM 中存在的主机状态。由于 Ansible 依赖于 idempotence，要使用 Ansible 将主机添加到 IdM，您必须创建一个 playbook，将主机的状态定义为 present: state: present。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：

- 您使用 Ansible 版本 2.14 或更高版本。
- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

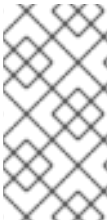
2. 创建一个 Ansible playbook 文件，其中包含您要确保的 IdM 中的 完全限定域名 (FQDN)。要使 Ansible playbook 为各个主机生成随机密码，即使主机已存在于 IdM 中，并且 update_password 设置为 on_create，请添加 random: yes 和 force: yes 选项。要简化此步骤，您可以复制 /usr/share/doc/ansible-freeipa/README-host.md Markdown 文件并对其进行相应的修改：

```
---
- name: Ensure hosts with random password
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Hosts host01.idm.example.com and host02.idm.example.com present with
random passwords
    ipahost:
      ipaadmin_password: "{{ ipaadmin_password }}"
      hosts:
      - name: host01.idm.example.com
        random: yes
        force: yes
      - name: host02.idm.example.com
```

```
    random: yes
    force: yes
  register: ipahost
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
are-present.yml
[...]
TASK [Hosts host01.idm.example.com and host02.idm.example.com present with
random passwords]
changed: [r8server.idm.example.com] => {"changed": true, "host":
{"host01.idm.example.com": {"randompassword": "0HoIRvjUdH0Ycbf6uYdWTxH"},
"host02.idm.example.com": {"randompassword": "5VdLgrf3wvojmACdHC3uA3s"}}}
```



**注意**

要使用随机、一次性密码(OTP)将主机部署为 IdM 客户端，请参阅使用 Ansible playbook 进行 IdM 客户端注册，或使用一次性密码安装客户端：交互式安装。

验证步骤

1. 以 admin 用户身份登录您的 IdM 服务器：

```
$ ssh admin@server.idm.example.com
Password:
```

2. 输入 ipa host-show 命令并指定其中一个主机的名称：

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Password: True
  Keytab: False
  Managed by: host01.idm.example.com
```

输出确认 IdM 中存在 host01.idm.example.com，并带有随机密码。

## 44.4. 使用 ANSIBLE PLAYBOOK 确保存在具有多个 IP 地址的 IDM 主机条目

按照以下流程，使用 Ansible playbook 确保主机条目在身份管理(IdM)中存在。主机条目通过其 完全限定域名 (FQDN)及其多个 IP 地址来定义。

**注意**

与 ipa host 实用程序相比，Ansible ipahost 模块可以确保主机存在或不存在多个 IPv4 和 IPv6 地址。ipa host-mod 命令无法处理 IP 地址。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

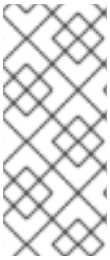
2. 创建 Ansible playbook 文件。将主机的 完全限定域名 (FQDN) 指定为 ipahost 变量的 name，用于确保主机的 IdM 中存在。使用 ip_address 语法在单独的行中指定多个 IPv4 和 IPv6 *ip_address* 值。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-

freeipa/playbooks/host/host-member-ipaddresses-present.yml 文件中的示例。您还可以包含附加信息：

```
---
- name: Host member IP addresses present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure host101.example.com IP addresses present
    ipahost:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: host01.idm.example.com
      ip_address:
      - 192.168.0.123
      - fe80::20c:29ff:fe02:a1b3
      - 192.168.0.124
      - fe80::20c:29ff:fe02:a1b4
      force: yes
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-host-
with-multiple-IP-addreses-is-present.yml
```



**注意**

这个过程在 IdM LDAP 服务器中创建主机条目，但不将主机注册到 IdM Kerberos 域。为此，您必须将主机部署为 IdM 客户端。详情请参阅使用 Ansible playbook 安装身份管理客户端。

验证步骤

1. 以 admin 用户身份登录您的 IdM 服务器：

```
$ ssh admin@server.idm.example.com
Password:
```

2. 输入 ipa host-show 命令并指定主机名称：

```
$ ipa host-show host01.idm.example.com
  Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
```

```
Password: False
Keytab: False
Managed by: host01.idm.example.com
```

输出确认 IdM 中存在 host01.idm.example.com。

3. 要验证 IdM DNS 记录中是否存在主机的多个 IP 地址，请输入 ipa dnsrecord-show 命令并指定以下信息：

- IdM 域的名称
- 主机的名称

```
$ ipa dnsrecord-show idm.example.com host01
[...]
  Record name: host01
  A record: 192.168.0.123, 192.168.0.124
  AAAA record: fe80::20c:29ff:fe02:a1b3, fe80::20c:29ff:fe02:a1b4
```

输出确认 playbook 中指定的所有 IPv4 和 IPv6 地址都已与 host01.idm.example.com 主机条目正确关联。

## 44.5. 使用 ANSIBLE PLAYBOOK 确保没有 IDM 主机条目

按照以下流程，使用 Ansible playbook 确保主机条目在身份管理(IdM)中不存在。

**先决条件**

- IdM 管理员凭证

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2.

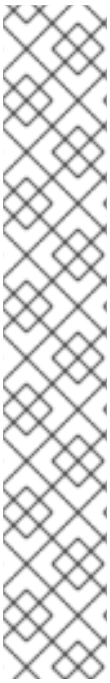
创建 Ansible playbook 文件，使其包含没有存在于 IdM 中的主机的完全限定域名 (FQDN)。如果您的 IdM 域集成了 DNS，请使用 updatedns: yes 选项从 DNS 中删除主机任意类型的关联记录。

要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/host/delete-host.yml 文件中的示例：

```
---
- name: Host absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Host host01.idm.example.com absent
    ipahost:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: host01.idm.example.com
      updatedns: yes
      state: absent
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-host-
absent.yml
```



注意

这个过程会产生：

- IdM Kerberos 域中没有的主机。
- IdM LDAP 服务器中不存在主机条目。

要从客户端主机本身中删除系统服务的特定 IdM 配置，如系统安全服务守护进程 (SSSD)，您必须在客户端上运行 ipa-client-install --uninstall 命令。详情请参阅卸载 IdM 客户端。

验证步骤

1. 以 admin 用户身份登录 ipaserver ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 显示 *host01.idm.example.com* 的信息：

```
$ ipa host-show host01.idm.example.com
ipa: ERROR: host01.idm.example.com: host not found
```

输出确认 IdM 中不存在该主机。

### 44.6. 其它资源

- 请参阅 /usr/share/doc/ansible-freeipa/README-host.md Markdown 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/host 目录中的其它 playbook。

# 第 45 章 使用 IDM CLI 管理主机组

了解如何使用以下操作在命令行界面(CLI)中管理主机组及其成员：

- 查看主机组及其成员
- 创建主机组
- 删除主机组
- 添加主机组成员
- 删除主机组成员
- 添加主机组成员管理者
- 删除主机组成员管理者

## 45.1. IDM 中的主机组

IdM 主机组可用于集中控制重要管理任务，特别是访问控制。

### 主机组的定义

主机组是包含一组具有通用访问控制规则和其他特征的 IdM 主机的实体。例如，您可以根据公司部门、物理位置或访问控制要求来定义主机组。

IdM 中的主机组可以包括：

- IdM 服务器和客户端

- 其他 IdM 主机组

默认创建的主机组

默认情况下，IdM 服务器为所有 IdM 服务器主机创建主机组 ipaservers。

直接和间接组成员

IdM 中的组属性同时适用于直接和间接成员：当主机组 B 是主机组 A 的成员时，主机组 B 的所有成员都被视为主机组 A 的间接成员。

## 45.2. 使用 CLI 查看 IDM 主机组

按照以下流程使用命令行界面(CLI)查看 IdM 主机组。

先决条件

- 管理 IdM 或用户管理员角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。

流程

1. 使用 ipa hostgroup-find 命令查找所有主机组。

```
$ ipa hostgroup-find
------------------
1 hostgroup matched
------------------
  Host-group: ipaservers
  Description: IPA server hosts
----------------------------
Number of entries returned 1
----------------------------
```

要显示主机组的所有属性，请添加 --all 选项。例如：

```
$ ipa hostgroup-find --all
------------------
```

```
1 hostgroup matched
------------------
  dn: cn=ipaservers,cn=hostgroups,cn=accounts,dc=idm,dc=local
  Host-group: ipaservers
  Description: IPA server hosts
  Member hosts: xxx.xxx.xxx.xxx
  ipauniqueid: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
  objectclass: top, groupOfNames, nestedGroup, ipaobject, ipahostgroup
----------------------------
Number of entries returned 1
----------------------------
```

### 45.3. 使用 CLI 创建 IDM 主机组

按照以下流程使用命令行界面(CLI)创建 IdM 主机组。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. 使用 ipa hostgroup-add 命令添加主机组。

例如，要创建名为 *group_name* 的 IdM 主机组，并为其提供描述：

```
$ ipa hostgroup-add --desc 'My new host group' group_name
---------------------
Added hostgroup "group_name"
---------------------
  Host-group: group_name
  Description: My new host group
---------------------
```

### 45.4. 使用 CLI 删除 IDM 主机组

按照以下流程使用命令行界面(CLI)删除 IdM 主机组。

**先决条件**

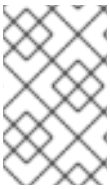
- 管理 IdM 或用户管理员角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。

**流程**

1. 使用 ipa hostgroup-del 命令删除主机组。

例如，要删除名为 *group_name* 的 IdM 主机组：

```
$ ipa hostgroup-del group_name
-------------------------
Deleted hostgroup "group_name"
-------------------------
```



**注意**

删除组不会从 IdM 中删除组成员。

## 45.5. 使用 CLI 添加 IDM 主机组成员

您可以使用单个命令，将主机和主机组作为成员添加到 IdM 主机组中。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。
- *可选*。使用 ipa hostgroup-find 命令查找主机和主机组。

**流程**

1. 要将成员添加到主机组，请使用 ipa hostgroup-add-member ，并提供相关信息。您可以使用这些选项指定要添加的成员类型：

- 使用 **--hosts** 选项，将一个或多个主机添加到 IdM 主机组。

例如，要将名为 ***example_member*** 的主机添加到名为 ***group_name*** 的组中：

```
$ ipa hostgroup-add-member group_name --hosts example_member
Host-group: group_name
Description: My host group
Member hosts: example_member
-------------------------
Number of members added 1
-------------------------
```

- 使用 **--hostgroups** 选项，将一个或多个主机组添加到 IdM 主机组。

例如，将名为 ***nested_group*** 的主机组添加到名为 ***group_name*** 的组中：

```
$ ipa hostgroup-add-member group_name --hostgroups nested_group
Host-group: group_name
Description: My host group
Member host-groups: nested_group
-------------------------
Number of members added 1
-------------------------
```

- 您可以使用以下语法在一个命令中将多个主机和多个主机组添加到 IdM 主机组中：

```
$ ipa hostgroup-add-member group_name --hosts={host1,host2} --hostgroups=
{group1,group2}
```

**重要**

将主机组添加为另一个主机组的成员时，请勿创建递归组。例如，如果组 **A** 是组 **B** 的成员，则不要将组 **B** 添加为组 **A** 的成员。递归组可能会导致无法预料的行为。

### 45.6. 使用 CLI 删除 IDM 主机组成员

您可以使用单个命令从 IdM 主机组中删除主机和主机组。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。

- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。

- *可选*。使用 ipa hostgroup-find 命令，确认该组中包含您要删除的成员。

**流程**

1. 要删除主机组成员，请使用 ipa hostgroup-remove-member 命令，并提供相关信息。您可以使用这些选项指定要删除的成员类型：

   - 使用 --hosts 选项从 IdM 主机组中删除一个或多个主机。
   例如，要从名为 *group_name* 的组中删除名为 *example_member* 的主机：

   ```
   $ ipa hostgroup-remove-member group_name --hosts example_member
   Host-group: group_name
   Description: My host group
   -------------------------
   Number of members removed 1
   -------------------------
   ```

   - 使用 --hostgroups 选项从 IdM 主机组中删除一个或多个主机组。
   例如，要从名为 *group_name* 的组中删除名为 *nested_group* 的主机组：

   ```
   $ ipa hostgroup-remove-member group_name --hostgroups example_member
   Host-group: group_name
   Description: My host group
   -------------------------
   Number of members removed 1
   -------------------------
   ```

**注意**

删除组不会从 IdM 中删除组成员。

- 您可以使用以下语法在一个命令中从 IdM 主机组中删除多个主机和多个主机组：

```
$ ipa hostgroup-remove-member group_name --hosts={host1,host2} --hostgroups=
{group1,group2}
```

### 45.7. 使用 CLI 添加 IDM 主机组成员管理者

您可以使用单个命令，将主机和主机组作为成员管理者添加到 IdM 主机组中。成员管理者可以将主机或主机组添加到 IdM 主机组，但不能更改主机组的属性。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。
- 您必须具有要添加为成员管理器的主机或主机组的名称，以及您要管理的主机组的名称。

**流程**

1. *可选*。使用 ipa hostgroup-find 命令查找主机和主机组。
2. 要将成员管理者添加到主机组，请使用 ipa hostgroup-add-member-manager。

   例如，将名为 *example_member* 的用户作为成员管理者添加到名为 *group_name* 的组中：

   ```
   $ ipa hostgroup-add-member-manager group_name --user example_member
   Host-group: group_name
   Member hosts: server.idm.example.com
   Member host-groups: project_admins
   Member of netgroups: group_name
   Membership managed by users: example_member
   -------------------------
   Number of members added 1
   -------------------------
   ```

3. 使用 --groups 选项，将一个或多个主机组作为成员管理者添加到 IdM 主机组中。

   例如，将名为 *admin_group* 的主机组作为成员管理者添加到名为 *group_name* 的组中：

   ```
   $ ipa hostgroup-add-member-manager group_name --groups admin_group
   Host-group: group_name
   Member hosts: server.idm.example.com
   Member host-groups: project_admins
   ```

```
Member of netgroups: group_name
Membership managed by groups: admin_group
Membership managed by users: example_member
-------------------------
Number of members added 1
-------------------------
```

**注意**

将成员管理者添加到主机组后，可能需要过些时间，才能将更新传播到身份管理环境中的所有客户端。

**验证步骤**

- 使用 ipa group-show 命令来验证主机用户和主机组被添加为成员管理者。

```
$ ipa hostgroup-show group_name
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
Membership managed by groups: admin_group
Membership managed by users: example_member
```

**其它资源**

- 如需了解更多详细信息，请参阅 ipa hostgroup-add-member-manager --help。
- 如需了解更多详细信息，请参阅 ipa hostgroup-show --help。

### 45.8. 使用 CLI 删除 IDM 主机组成员管理者

您可以使用单个 命令，将主机和主机组作为成员管理者从 IdM 主机组中删除。成员管理者可以从 IdM 主机组中删除主机组成员管理者，但不能更改主机组的属性。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 一个活跃的 Kerberos 票据。详情请参阅 使用 kinit 手动登录到 IdM。

- 您必须具有要删除的现有成员管理者主机组的名称，以及它们正在管理的主机组的名称。

**流程**

1. *可选*。使用 **ipa hostgroup-find** 命令查找主机和主机组。

2. 要从主机组中删除成员管理者，请使用 **ipa hostgroup-remove-member-manager** 命令。

   例如，要从名为 ***group_name*** 的组中删除作为成员管理者的名为 ***example_member*** 的用户：

```
$ ipa hostgroup-remove-member-manager group_name --user example_member
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
Member of netgroups: group_name
Membership managed by groups: nested_group
-------------------------
Number of members removed 1
-------------------------
```

3. 使用 **--groups** 选项，将一个或多个主机组作为成员管理者从 IdM 主机组中删除。

   例如，要从名为 ***group_name*** 的组中删除作为成员管理者的名为 ***nested_group*** 的主机组：

```
$ ipa hostgroup-remove-member-manager group_name --groups nested_group
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
Member of netgroups: group_name
-------------------------
Number of members removed 1
-------------------------
```

**注意**

从主机组中删除成员管理者后，可能需要过些时间，才能将更新传播到身份管理环境中的所有客户端。

验证步骤

- 使用 ipa group-show 命令来验证主机用户和主机组已作为成员管理者被删除。

```
$ ipa hostgroup-show group_name
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
```

**其它资源**

- 如需了解更多详细信息，请参阅 ipa hostgroup-remove-member-manager --help。
- 如需了解更多详细信息，请参阅 ipa hostgroup-show --help。

# 第 46 章 使用 IDM WEB UI 管理主机组

了解如何使用以下操作在 Web 界面(Web UI)中管理主机组及其成员：

- 查看主机组及其成员
- 创建主机组
- 删除主机组
- 添加主机组成员
- 删除主机组成员
- 添加主机组成员管理者
- 删除主机组成员管理者

## 46.1. IDM 中的主机组

IdM 主机组可用于集中控制重要管理任务，特别是访问控制。

### 主机组的定义

主机组是包含一组具有通用访问控制规则和其他特征的 IdM 主机的实体。例如，您可以根据公司部门、物理位置或访问控制要求来定义主机组。

IdM 中的主机组可以包括：

- IdM 服务器和客户端

- 其他 IdM 主机组

默认创建的主机组

默认情况下，IdM 服务器为所有 IdM 服务器主机创建主机组 ipaservers。

直接和间接组成员

IdM 中的组属性同时适用于直接和间接成员：当主机组 B 是主机组 A 的成员时，主机组 B 的所有成员都被视为主机组 A 的间接成员。

## 46.2. 在 IDM WEB UI 中查看主机组

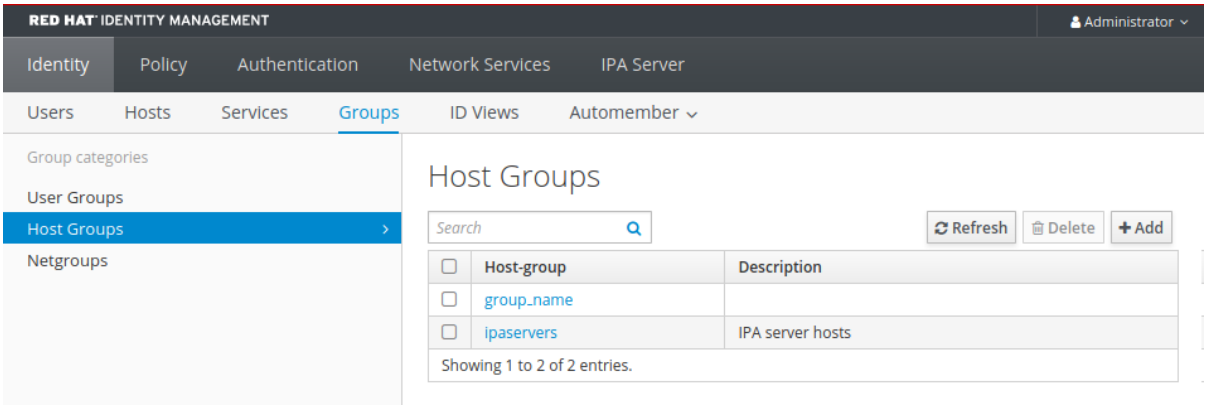
按照以下流程，使用 Web 界面(Web UI)查看 IdM 主机组。

先决条件

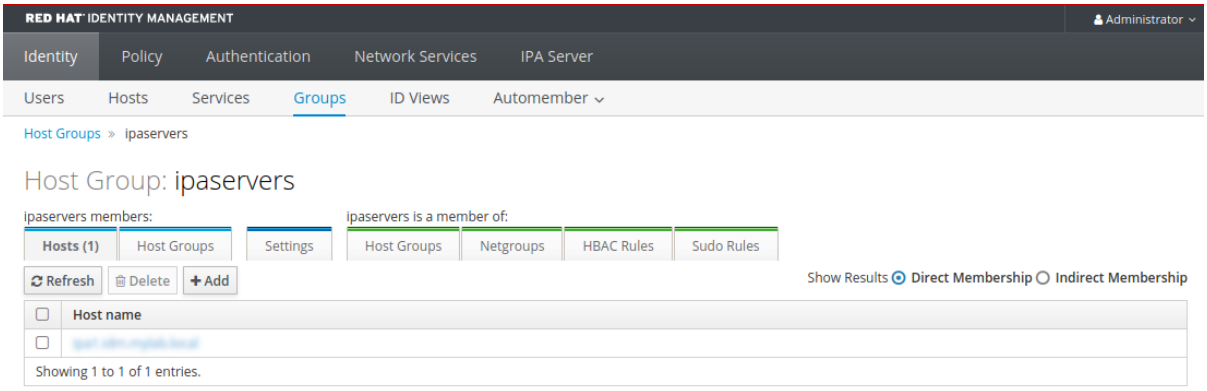
- 管理 IdM 或用户管理员角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

流程

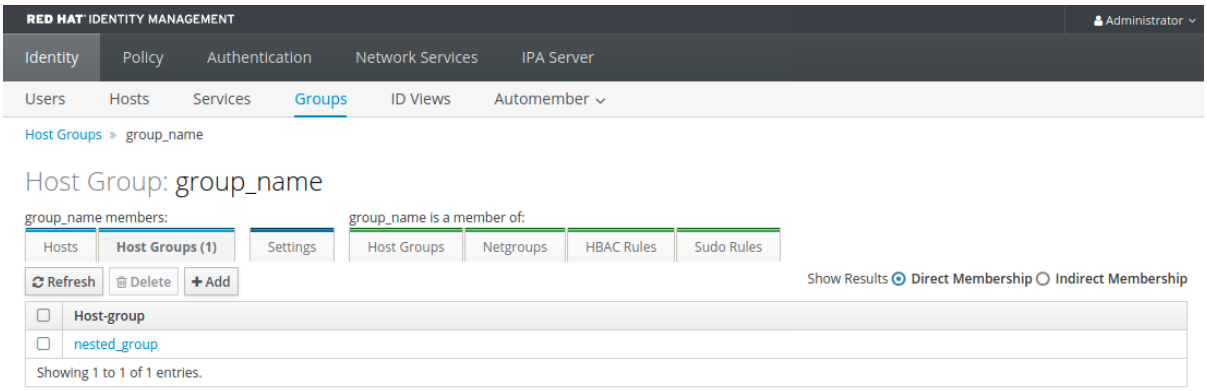
1. 点击 Identity → Groups，然后选择 Host Groups 选项卡。
   - 页面中列出了现有的主机组及其描述。
   - 您可以搜索特定的主机组。




2. **单击列表中的组，来显示属于此组的主机。您可以将结果限制为直接或间接的成员。**




3. **选择 Host Groups 选项卡，来显示属于此组的主机组（嵌套主机组）。您可以将结果限制为直接或间接的成员。**




## 46.3. 在 IDM WEB UI 中创建主机组

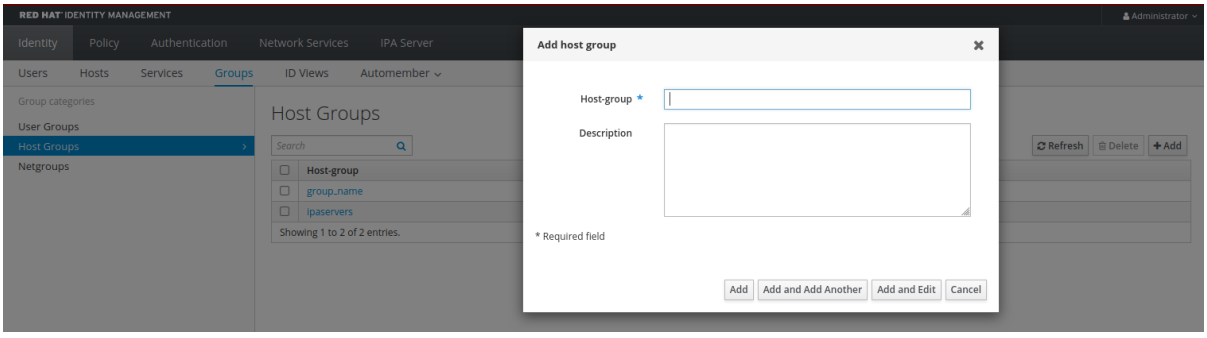
按照以下流程，使用 Web 界面(Web UI)创建 IdM 主机组。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

**流程**

1. 点击 Identity → Groups，然后选择 Host Groups 选项卡。
2. 点击 Add。此时出现 Add host grou 对话框。
3. 提供有关组的信息：name（必需的）和 description（可选的）。
4. 单击 Add 确认。




## 46.4. 在 IDM WEB UI 中删除主机组

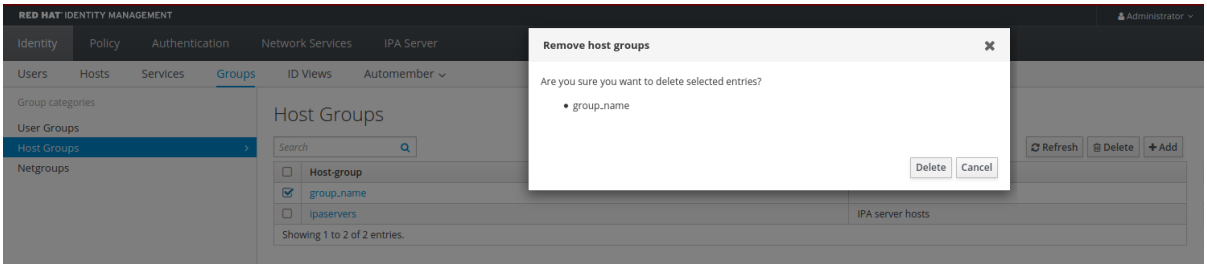
按照以下流程，使用 Web 界面(Web UI)删除 IdM 主机组。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

**流程**

1. 点击 Identity → Groups，并选择 Host Groups 选项卡。
2. 选择要删除的 IdM 主机组，单击 Delete。此时会出现确认对话框。
3. 单击 Delete 确认




**注意**

删除主机组不会从 IdM 中删除组成员。

## 46.5. 在 IDM WEB UI 中添加主机组成员

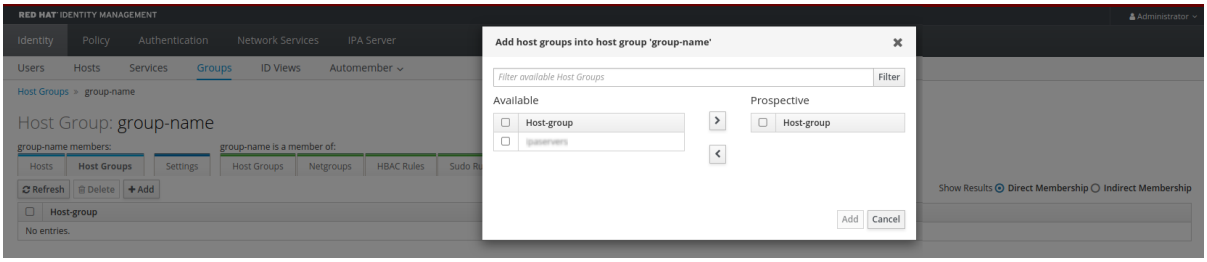
按照以下流程，使用 Web 界面(Web UI)在 IdM 中添加主机组成员。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

**流程**

1. 点击 **Identity → Groups**，并选择 **Host Groups** 选项卡。
2. 单击您要添加成员的组的名称。
3. 单击 **Hosts** 或 **Host groups** 选项卡，具体取决于您要添加的成员的类型。此时会出现相应的对话框。
4. 选择要添加的主机或主机组，然后点击 **>** 箭头按钮将它们移到 **Prospective** 列中。
5. 单击 **Add** 确认。




## 46.6. 在 IDM WEB UI 中删除主机组成员

按照以下流程，使用 Web 界面(Web UI)删除 IdM 中的主机组成员。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

**流程**

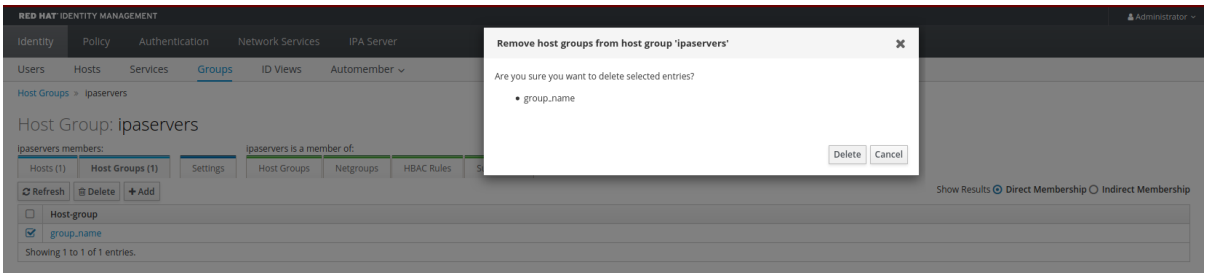
1. 点击 **Identity → Groups**，并选择 **Host Groups** 选项卡。

2. 单击您要从中删除成员的组的名称。

3. 单击 **Hosts** 或 **Host groups** 选项卡，具体取决于您要删除的成员的类型。

4. 选中您要删除的成员旁边的复选框。

5. 点击 **Delete**。此时会出现确认对话框。




6. 单击 **Delete** 确认。已选择的成员被删除。

## 46.7. 使用 WEB UI 添加 IDM 主机组成员管理者

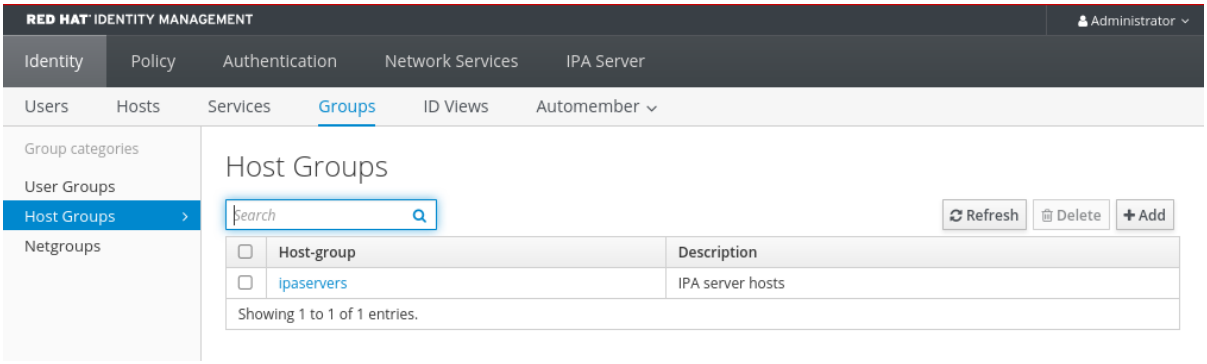
**按照以下流程，使用 Web 界面(Web UI)将用户或用户组作为主机组成员管理者添加到 IdM 中。成员管理者可以将主机组成员管理者添加到 IdM 主机组中，但不能更改主机组的属性。**

**先决条件**

- **管理 IdM 或用户管理员角色的管理员特权。**
- **您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。**
- **您必须有要添加为成员管理者的主机组的名称，以及您要管理的主机组的名称。**

**流程**

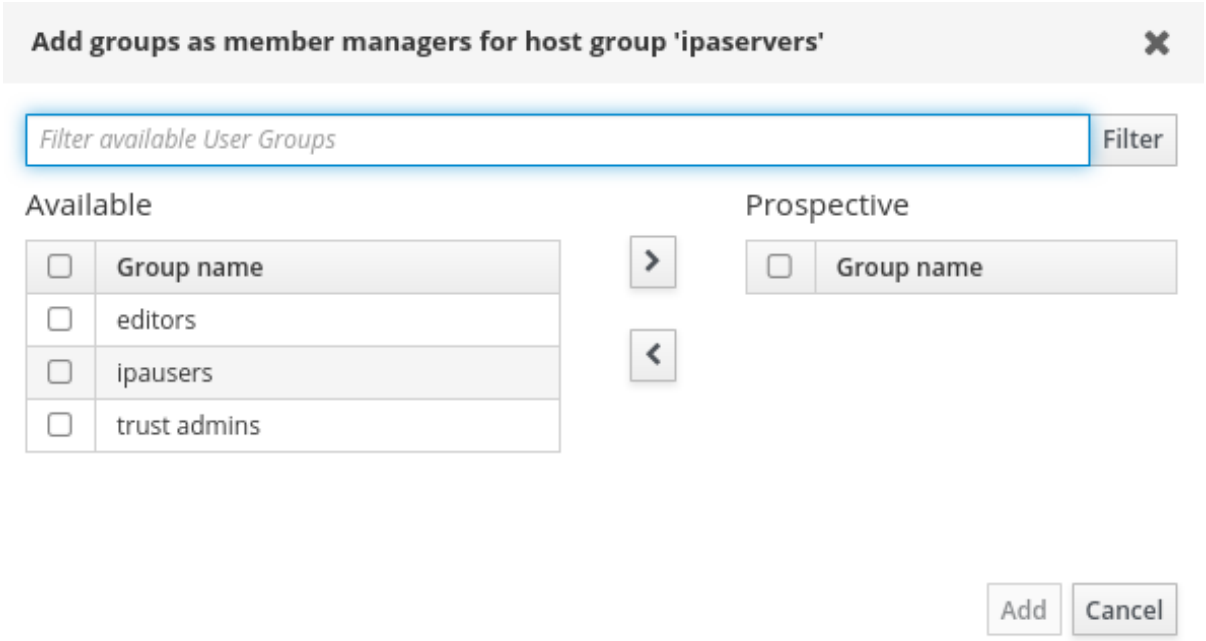
1. 点击 **Identity → Groups，并选择 Host Groups** 选项卡。




2. 单击要添加成员管理者的组的名称。

3. 单击 **member managers** 选项卡 **User Groups or Users**，具体取决于您要添加的成员管理者的类型。此时会出现相应的对话框。

4. 点击 **Add**。




5. 选择要添加的用户或用户组，然后点击 **>** 箭头按钮，将它们移到 **Prospective** 列中。

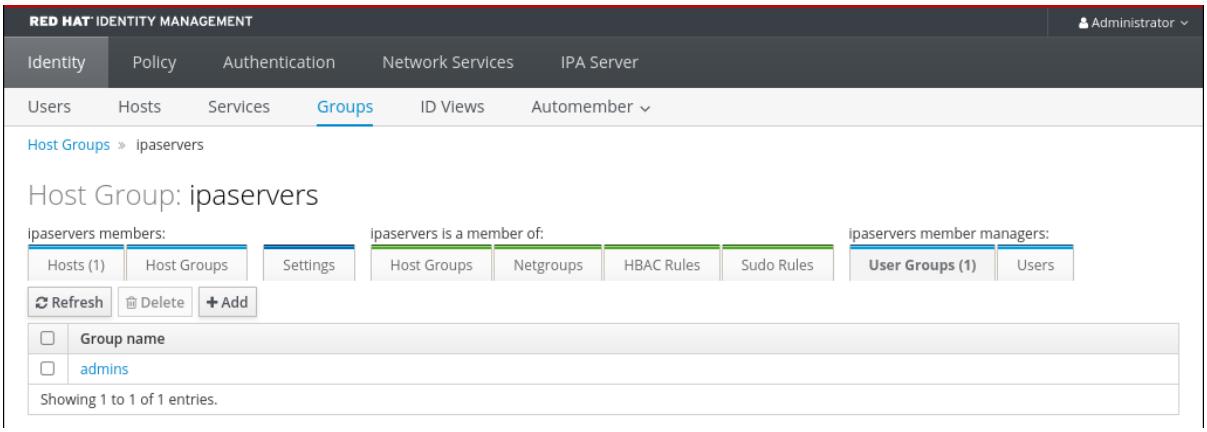
6. 单击 **Add** 确认。

**注意**

将成员管理者添加到主机组后，可能需要过些时间，才能将更新传播到身份管理环境中的所有客户端。

**验证步骤**

- 在主机组对话框中，验证用户组或用户已被添加到组或用户的成员管理者列表中。




## 46.8. 使用 WEB UI 删除 IDM 主机组成员管理者

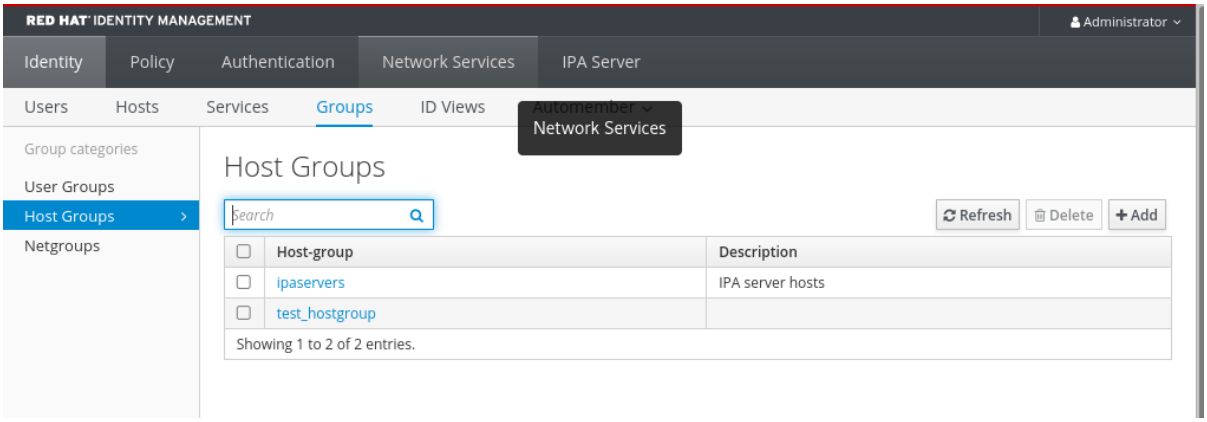
按照以下流程，使用 Web 界面(Web UI)删除 IdM 中作为主机组成员管理者的用户或用户组。成员管理者可以从 IdM 主机组中删除主机组成员管理者，但不能更改主机组的属性。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。
- 您必须具有要删除的现有成员管理者主机组的名称，以及它们正在管理的主机组的名称。

**流程**

1. 点击 **Identity** → **Groups**，并选择 **Host Groups** 选项卡。

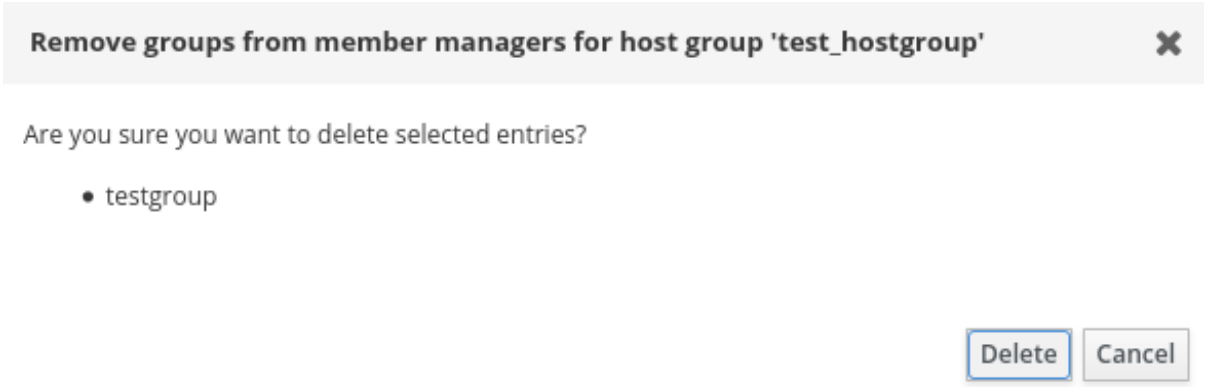



2. 单击您要从中删除成员管理者的组的名称。

3. 单击 **member managers** 选项卡 **User Groups** 或**Users**，具体取决于您要删除的成员管理者的类型。此时会出现相应的对话框。

4. 选择要删除的用户或用户组，然后单击 **Delete**。

5. 单击 **Delete** 确认。

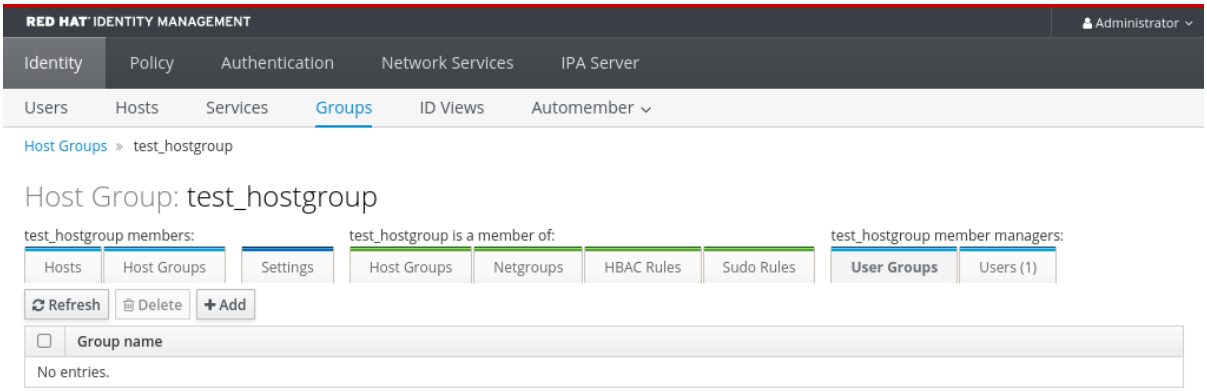



**注意**

从主机组中删除成员管理者后，可能需要过些时间，才能将更新传播到身份管理环境中的所有客户端。

验证步骤

- 在主机组对话框中，验证用户组或用户已从组或用户的成员管理者列表中删除。


RED HAT IDENTITY MANAGEMENT
Administrator
Identity
Policy
Authentication
Network Services
IPA Server
Users
Hosts
Services
Groups
ID Views
Automember
Host Groups » test_hostgroup
Host Group: test_hostgroup
test_hostgroup members:
Hosts
Host Groups
Settings
test_hostgroup is a member of:
Host Groups
Netgroups
HBAC Rules
Sudo Rules
test_hostgroup member managers:
User Groups
Users (1)
Refresh
Delete
Add
Group name
No entries.

# 第 47 章 使用 ANSIBLE PLAYBOOK 管理主机组

要了解更多有关 身份管理(IdM)中主机组 的信息，并使用 Ansible 来执行涉及身份管理(IdM)中主机组的操作，请参阅：

- IdM 中的主机组
- 确保存在 IdM 主机组
- 确保 IdM 主机组中存在主机
- 嵌套 IdM 主机组
- 确保 IdM 主机组中存在成员管理器
- 确保 IdM 主机组中没有主机
- 确保 IdM 主机组没有嵌套的主机组
- 确保 IdM 主机组中没有成员管理器

## 47.1. IDM 中的主机组

IdM 主机组可用于集中控制重要管理任务，特别是访问控制。

### 主机组的定义

主机组是包含一组具有通用访问控制规则和其他特征的 IdM 主机的实体。例如，您可以根据公司部门、物理位置或访问控制要求来定义主机组。

IdM 中的主机组可以包括：

- IdM 服务器和客户端
- 其他 IdM 主机组

默认创建的主机组

默认情况下，IdM 服务器为所有 IdM 服务器主机创建主机组 ipaservers。

直接和间接组成员

IdM 中的组属性同时适用于直接和间接成员：当主机组 B 是主机组 A 的成员时，主机组 B 的所有成员都被视为主机组 A 的间接成员。

## 47.2. 使用 ANSIBLE PLAYBOOK 确保存在 IDM 主机组

按照以下流程，使用 Ansible playbook 确保在主机组在身份管理(IdM)中存在。

注意

如果没有 Ansible，则使用 ipa hostgroup-add 命令在 IdM 中创建主机组条目。将主机组添加到 IdM 的结果是 IdM 中存在主机组的状态。由于 Ansible 依赖幂等性，要使用 Ansible 将主机组添加到 IdM，您必须创建一个 playbook，其中将主机组的状态定义为 present: state: present。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名

(FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

流程

1. 创建一个清单文件，如 inventory.file，并使用目标 IdM 服务器列表定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的主机组信息，创建 Ansible playbook 文件。例如，若要确保存在名为 databases 的主机组，可在 - ipahostgroup 任务中指定 name: databases。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-present.yml 文件中的示例。

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure host-group databases is present
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      state: present
```

在 playbook 中， state: present 表示将主机组添加到 IdM 的请求，除非该主机组在那里已存在。

3. 运行 playbook ：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
hostgroup-is-present.yml
```

验证步骤

1. 以 admin 用户身份登录 ipaserver ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 为 admin 请求一个 Kerberos ticket：

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. 显示在 IdM 中存在的主机组的信息，以确保：

```
$ ipa hostgroup-show databases
  Host-group: databases
```

IdM 中存在 databases 主机组。

### 47.3. 确保使用 ANSIBLE PLAYBOOK 在 IDM 主机组中存在主机

按照以下流程，使用 Ansible playbook 确保主机在身份管理(IdM)中的主机组中存在。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- IdM 中已存在您要引用的主机。详情请参阅使用 Ansible playbook 确保存在 IdM 主机条目。

- 您从 Ansible playbook 文件中引用的主机组已添加到 IdM 中。详情请参阅确保使用 Ansible playbook 确保 IdM 主机组存在。

**流程**

1. 创建一个清单文件，如 inventory.file，并使用目标 IdM 服务器列表定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的主机信息，创建 Ansible playbook 文件。使用 ipahostgroup 变量的 name 参数，指定主机组的名称。使用 ipahostgroup 变量的 host 参数指定主机名称。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-in-hostgroup.yml 文件中的示例：

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure host-group databases is present
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      host:
      - db.idm.example.com
      action: member
```

此 playbook 将 db.idm.example.com 主机添加到 databases 主机组。action: member 行表示在 playbook 运行时，不会尝试添加 databases 组本身。相反，只尝试将 db.idm.example.com 添加到数据库。

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
or-hostgroups-are-present-in-hostgroup.yml
```

验证步骤

1. 以 admin 用户身份登录 ipaserver ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 为 admin 请求一个 Kerberos ticket：

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. 显示主机组的信息以查看其中存在哪些主机：

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member hosts: db.idm.example.com
```

db.idm.example.com 主机显示为 databases 主机组的成员。

### 47.4. 使用 ANSIBLE PLAYBOOK 嵌套 IDM 主机组

按照以下流程，使用 Ansible playbook 确保嵌套的主机组在身份管理(IdM)主机组中存在。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：

- 您使用 Ansible 版本 2.14 或更高版本。
- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- IdM 中已存在您从 Ansible playbook 文件中引用的主机组。详情请参阅确保使用 Ansible playbook 确保 IdM 主机组存在。

**流程**

1. 创建一个清单文件，如 inventory.file，并使用目标 IdM 服务器列表定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的主机组信息，创建 Ansible playbook 文件。为确保嵌套的主机组 ***A*** 存在于主机组 ***B*** 中：在 Ansible playbook 的 - ipahostgroup 变量中使用 name 变量指定主机组 ***B*** 的名称。使用 hostgroup 变量指定嵌套主机组 ***A*** 的名称。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-in-hostgroup.yml 文件中的示例：

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure hosts and hostgroups are present in existing databases hostgroup
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      hostgroup:
```

```
    - mysql-server
    - oracle-server
    action: member
```

此 Ansible playbook 确保在 databases 主机组中存在 myqsl-server 和 oracle-server 主机组。action: member 行表示在 playbook 运行时，不会尝试将 databases 组本身添加到 IdM。

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
or-hostgroups-are-present-in-hostgroup.yml
```

验证步骤

1. 以 admin 用户身份登录 ipaserver ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 为 admin 请求一个 Kerberos ticket：

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. 显示有关存在嵌套主机组的主机组的信息：

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member hosts: db.idm.example.com
  Member host-groups: mysql-server, oracle-server
```

mysql-server 和 oracle-server 主机组存在于 databases 主机组中。

## 47.5. 使用 ANSIBLE PLAYBOOK 在 IDM 主机组中存在成员管理器

以下流程描述了确保使用 Ansible playbook 在 IdM 主机和主机组中存在成员管理器。

先决条件

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您必须具有要添加为成员管理器的主机或主机组的名称，以及您要管理的主机组的名称。

流程

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的主机和主机组成员管理信息创建一个 Ansible playbook 文件：

```
---

- name: Playbook to handle host group membership management
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
```

```
- name: Ensure member manager user example_member is present for group_name
  ipahostgroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: group_name
    membermanager_user: example_member

- name: Ensure member manager group project_admins is present for group_name
  ipahostgroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: group_name
    membermanager_group: project_admins
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-member-
managers-host-groups.yml
```

**验证步骤**

您可以使用 ipa group-show 命令验证 group_name 组是否包含 example_member 和project_admins 作为成员管理者：

1. 以管理员身份登录到 ipaserver：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 显示有关 *testhostgroup* 的信息：

```
ipaserver]$ ipa hostgroup-show group_name
  Host-group: group_name
  Member hosts: server.idm.example.com
  Member host-groups: testhostgroup2
  Membership managed by groups: project_admins
  Membership managed by users: example_member
```

**其它资源**

- 请参阅 ipa hostgroup-add-member-manager --help。
- 请参阅 ipa man page。

## 47.6. 使用 ANSIBLE PLAYBOOK 确保 IDM 主机组中没有主机

按照以下流程，使用 Ansible playbook 确保主机组中的主机在身份管理(IdM)中不存在。

### 先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- IdM 中已存在您要引用的主机。详情请参阅使用 Ansible playbook 确保存在 IdM 主机条目。
- IdM 中已存在您从 Ansible playbook 文件中引用的主机组。详情请参阅确保使用 Ansible playbook 确保 IdM 主机组存在。

### 流程

1. 创建一个清单文件，如 inventory.file，并使用目标 IdM 服务器列表定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 创建含有必要的主机和主机组信息的 Ansible playbook 文件。使用 ipahostgroup 变量的 name 参数，指定主机组的名称。使用 ipahostgroup 变量的 host 参数指定要确保其不存在于主机组中的主机名称。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-in-hostgroup.yml 文件中的示例：

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure host-group databases is absent
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      host:
      - db.idm.example.com
      action: member
      state: absent
```

此 playbook 确保 db.idm.example.com 主机没有存在于 databases 主机组中。action: member 行表示在 playbook 运行时，不会尝试删除 databases 组本身。

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
or-hostgroups-are-absent-in-hostgroup.yml
```

验证步骤

1. 以 admin 用户身份登录 ipaserver ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 为 admin 请求一个 Kerberos ticket：

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. 显示主机组及其包含的主机的信息：

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member host-groups: mysql-server, oracle-server
```

在 databases 主机组中不存在 db.idm.example.com 主机。

## 47.7. 使用 ANSIBLE PLAYBOOK 确保 IDM 主机组没有嵌套的主机组

按照以下流程，使用 Ansible playbook 确保来自外部主机组的嵌套的主机组在身份管理(IdM)中不存在。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- IdM 中已存在您从 Ansible playbook 文件中引用的主机组。详情请参阅确保使用 Ansible playbook 确保 IdM 主机组存在。

**流程**

1. 创建一个清单文件，如 inventory.file，并使用目标 IdM 服务器列表定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的主机组信息，创建 Ansible playbook 文件。在 - ipahostgroup 变量中使用 name 变量指定外部主机组的名称。使用 hostgroup 变量指定嵌套主机组的名称。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-in-hostgroup.yml 文件中的示例：

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure hosts and hostgroups are absent in existing databases hostgroup
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      hostgroup:
      - mysql-server
      - oracle-server
      action: member
      state: absent
```

此 playbook 确保 mysql-server 和 oracle-server 主机组没有存在于 databases 主机组中。action: member 行表示，在 playbook 运行时，不会尝试确保从 IdM 中删除 databases 组本身。

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
or-hostgroups-are-absent-in-hostgroup.yml
```

验证步骤

1. 以 admin 用户身份登录 ipaserver ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 为 admin 请求一个 Kerberos ticket：

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. 显示应当缺少嵌套主机组的主机组的信息：

```
$ ipa hostgroup-show databases
  Host-group: databases
```

输出确认，外部 databases 主机组中没有mysql-server 和 oracle-server 嵌套式主机组。

## 47.8. 使用 ANSIBLE PLAYBOOK 确保没有 IDM 主机组

按照以下流程，使用 Ansible playbook 确保主机组在身份管理(IdM)中不存在。

**注意**

如果没有 Ansible，则使用 ipa hostgroup-del 命令从 IdM 中删除主机组条目。从 IdM 中删除主机组的结果是 IdM 中缺少主机组的状态。由于 Ansible 依赖于 idempotence，若要使用 Ansible 从 IdM 中删除主机组，您必须创建一个 playbook，它将主机组的状态定义为 absent: state: absent。

**先决条件**

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。

- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 创建一个清单文件，如 inventory.file，并使用目标 IdM 服务器列表定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的主机组信息，创建 Ansible playbook 文件。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-absent.yml 文件中的示例。

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - Ensure host-group databases is absent
    ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      state: absent
```

此 playbook 确保 IdM 中没有 databases 主机组。state: absent 表示从 IdM 中删除主机组的请求，除非它已被删除。

3. 运行 playbook ：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
hostgroup-is-absent.yml
```

验证步骤

1. 以 admin 用户身份登录 ipaserver ：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 为 admin 请求一个 Kerberos ticket：

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. 显示您没有保证的主机组的信息：

```
$ ipa hostgroup-show databases
ipa: ERROR: databases: host group not found
```

IdM 中不存在 databases 主机组。

## 47.9. 使用 ANSIBLE PLAYBOOK 确保 IDM 主机组中没有成员管理器

以下流程描述了确保使用 Ansible playbook 在 IdM 主机和主机组中存在成员管理器。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 您必须具有要作为成员管理者删除的用户或用户组的名称，以及它们所管理的主机组的名称。

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 使用必要的主机和主机组成员管理信息创建一个 Ansible playbook 文件：

```
---

- name: Playbook to handle host group membership management
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure member manager host and host group members are absent for
group_name
    ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: group_name
      membermanager_user: example_member
      membermanager_group: project_admins
      action: member
      state: absent
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
member-managers-host-groups-are-absent.yml
```

验证步骤

您可以使用 ipa group-show 命令验证 group_name 组不包含 example_member 或 project_admins 作为成员管理者：

1. 以管理员身份登录到 ipaserver：

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. 显示有关 *testhostgroup* 的信息：

```
ipaserver]$ ipa hostgroup-show group_name
  Host-group: group_name
  Member hosts: server.idm.example.com
  Member host-groups: testhostgroup2
```

**其它资源**

- 请参阅 ipa hostgroup-add-member-manager --help。
- 请参阅 ipa man page。

# 第 48 章 为用户、主机和服务管理 KERBEROS 主体别名

当您创建新用户、主机或服务时，会自动添加以下格式的 Kerberos 主体：

- *user_name@REALM*
- host/*host_name@REALM*
- *service_name/host_name@REALM*

管理员可以让用户、主机或服务使用别名对 Kerberos 应用进行身份验证。这在以下情况下很有用：

- 用户名已更改，用户希望使用之前的用户名和新用户名登录。
- 即使 IdM Kerberos 域与电子邮件域不同，用户也需要使用电子邮件地址登录。

请注意，如果您重命名了用户，对象会保留别名和之前的规范主体名称。

## 48.1. 添加一个 KERBEROS 主体别名

您可以在身份管理(IdM)环境中将别名名称与现有 Kerberos 主体关联。这增强了安全性，并简化了 IdM 域中的身份验证过程。

**流程**

- 要将别名名称 useralias 添加到帐户 user 中，请输入：

```
# ipa user-add-principal <user> <useralias>
-------------------------------
Added new aliases to user "user"
-------------------------------
        User login: user
  Principal alias: user@IDM.EXAMPLE.COM, useralias@IDM.EXAMPLE.COM
```

要为主机或服务添加一个别名，请分别使用 ipa host-add-principal 或 ipa service-add-principal 命令。

如果您使用别名名称进行身份验证，请使用 kinit 命令的 -C 选项：

```
# kinit -C <useralias>
Password for <user>@IDM.EXAMPLE.COM:
```

## 48.2. 删除一个 KERBEROS 主体别名

您可以在其身份管理(IdM)环境中删除与 Kerberos 主体关联的别名名称。

**流程**

- 要从帐户 user 中删除别名 useralias，请输入：

```
# ipa user-remove-principal <user> <useralias>
-------------------------------
Removed aliases from user "user"
-------------------------------
  User login: user
  Principal alias: user@IDM.EXAMPLE.COM
```

要从主机或服务中删除一个别名，请分别使用 ipa host-remove-principal 或 ipa service-remove-principal 命令。

请注意，您无法删除规范主体名称：

```
# ipa user-show <user>
  User login: user
  ...
  Principal name: user@IDM.EXAMPLE.COM
  ...

# ipa user-remove-principal user user
ipa: ERROR: invalid 'krbprincipalname': at least one value equal to the canonical principal name must be present
```

## 48.3. 添加一个 KERBEROS 企业主体别名

您可以在身份管理(IdM)环境中将企业级别名名称与现有 Kerberos 企业主体关联。企业主体别名可以

使用任何域后缀，但用户主体名称(UPN)后缀、NetBIOS 名称或可信活动目录林域的域名除外。

注意

在添加或删除企业级别名时，请使用两个反斜杠(\\)转义 @ 符号。否则，shell 将 @ 符号解释为 Kerberos 域名称的一部分，并导致以下错误：

```
ipa: ERROR: The realm for the principal does not match the realm for this IPA server
```

流程

- 将企业主体别名 user@example.com 添加到 user 帐户中：

```
# ipa user-add-principal <user> <user\\@example.com>
------------------------------
Added new aliases to user "user"
------------------------------
        User login: user
   Principal alias: user@IDM.EXAMPLE.COM,
user\@example.com@IDM.EXAMPLE.COM
```

要向主机或服务添加一个企业别名，请分别使用 ipa host-add-principal 或 ipa service-add-principal 命令。

如果您使用企业主体名称进行身份验证，请使用 kinit 命令的 -E 选项：

```
# kinit -E <user@example.com>
Password for user\@example.com@IDM.EXAMPLE.COM:
```

### 48.4. 删除 KERBEROS 企业主体别名

您可以在其身份管理(IdM)环境中删除与 Kerberos 企业主体关联的企业别名名称。

注意

在添加或删除企业级别名时，请使用两个反斜杠(\\)转义 @ 符号。否则，shell 将 @ 符号解释为 Kerberos 域名称的一部分，并导致以下错误：

```
ipa: ERROR: The realm for the principal does not match the realm for this IPA server
```

**流程**

- 要从帐户 user 中删除企业主体别名 user@example.com，请输入：

```
# ipa user-remove-principal <user> <user\\@example.com>
-------------------------------
Removed aliases from user "user"
-------------------------------
  User login: user
  Principal alias: user@IDM.EXAMPLE.COM
```

要从主机或服务中删除一个别名，请分别使用 ipa host-remove-principal 或 ipa service-remove-principal 命令。

# 第 49 章 管理 KERBEROS 标记

Kerberos 标志对于在支持 Kerberos 的网络环境中指定身份验证机制、授权级别和安全协议至关重要。使用 Kerberos 标志，您可以确保安全访问控制，防止未经授权的访问，并改进不同 Kerberos 实现之间的互操作性。

## 49.1. 服务和主机的 KERBEROS 标志

您可以使用各种 Kerberos 标志来定义 Kerberos 票据行为的特定方面。您可以将这些标志添加到服务和主机 Kerberos 主体。

身份管理(IdM)中的主体接受以下 Kerberos 标记：

- **OK_AS_DELEGATE**

  使用此标志指定可委托的 Kerberos 票据。

  Active Directory (AD)客户端检查 Kerberos 票据上的 OK_AS_DELEGATE 标志，以确定用户凭证是否可以转发或委派给特定的服务器。AD 仅将票据授予票据(TGT)转发到配置了 OK_AS_DELEGATE 的服务或主机。使用这个标志，系统安全服务守护进程(SSSD)可以将 AD 用户 TGT 添加到 IdM 客户端机器上的默认 Kerberos 凭证缓存中。

- **REQUIRES_PRE_AUTH**

  使用此标志指定只允许预先验证的票据对主体进行身份验证。

  设置 REQUIRES_PRE_AUTH 标志后，密钥分发中心(KDC)需要额外的身份验证：只有 TGT 已被预先验证，则 KDC 会发出带有 REQUIRES_PRE_AUTH 的主体的 TGT。

  您可以清除 REQUIRES_PRE_AUTH，以禁用所选服务或主机的预身份验证。这降低了 KDC 上的负载，但稍微增加对长期密钥的 brute-force 攻击的可能性。

- **OK_TO_AUTH_AS_DELEGATE**

  使用 OK_TO_AUTH_AS_DELEGATE 标志指定允许该服务代表用户获取 Kerberos 票据。

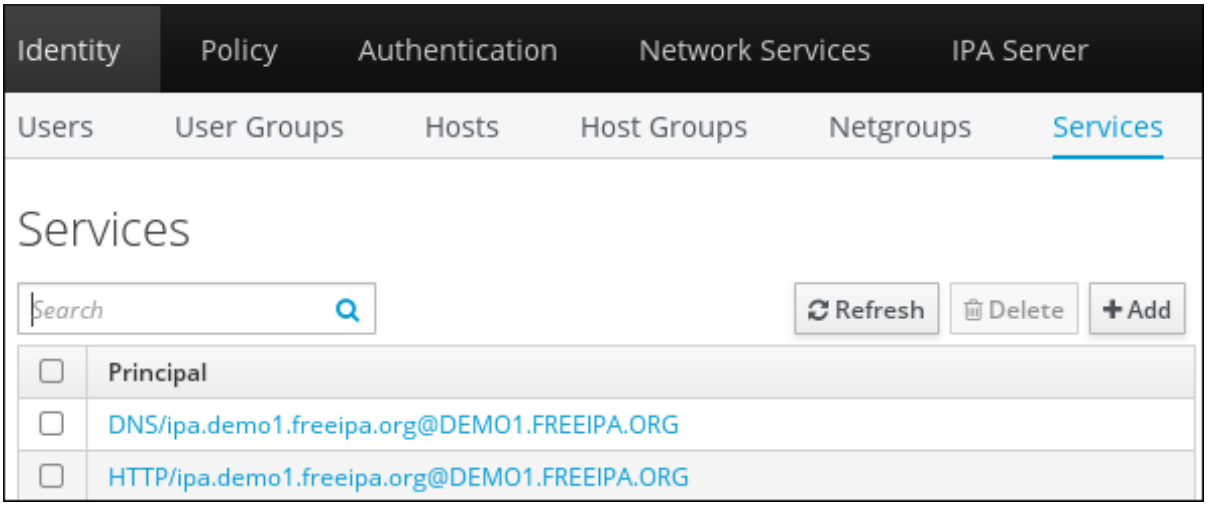
请注意，为了代表用户获取其他票据，该服务需要 OK_AS_DELEGATE 标志以及密钥分发方允许的对应策略决定。

## 49.2. 从 WEB UI 设置 KERBEROS 标志

您可以使用 IdM Web UI 设置 Kerberos 标志。以下流程将 Kerberos 标志设置为主体。

**流程**

1. 在菜单中选择 **Identity** → **Services**。




2. 点击您要向其添加标志的服务。

3. 检查您要设置的选项：

   - 要设置 OK_AS_DELEGATE 标志，请选中 **Trusted for delegation**。
   - 要设置 REQUIRES_PRE_AUTH 标志，请检查 **Requires pre-authentication**。
   - 要设置 OK_TO_AUTH_AS_DELEGATE 标志，请选中 **Trusted 以用户 进行身份验证**。

### 49.3. 从命令行设置和删除 KERBEROS 标志

您可以使用命令行添加或删除 Kerberos 标记。ipa service-mod 命令对标志使用以下命令选项：

- OK_AS_DELEGATE 的 --ok-as-delegate
- REQUIRES_PRE_AUTH 的 --requires-pre-auth
- --OK-to-auth-as-delegate for OK_TO_AUTH_AS_DELEGATE

通过将选项值设置为 1，您可以为原则启用标记。通过将选项值设为 0，您可以禁用 标志。

以下流程为 服务/ipa.example.com@example.com 主体启用和禁用 OK_AS_DELEGATE 标志。

流程

- 要为 服务/ipa.example.com@example.com 原则添加 OK_AS_DELEGATE 标志，请运行：

```
$ ipa service-mod service/ipa.example.com@EXAMPLE.COM --ok-as-delegate=1
```

- 要从 服务/ipa.example.com@example.com 原则中删除 OK_AS_DELEGATE 标志，请运行：

```
$ ipa service-mod service/ipa.example.com@EXAMPLE.COM --ok-as-delegate=0
```

### 49.4. 从命令行显示 KERBEROS 标志

您可以使用命令行显示 Kerberos 标记设置。以下流程显示 demo/ipa.example.com@EXAMPLE.COM 主体的 OK_AS_DELEGATE 标志。

流程

要找出是否为主体设置了 OK_AS_DELEGATE ：

1. 运行 kvno 工具：

```
$ kvno demo/ipa.example.com@EXAMPLE.COM
```

2. 要显示标志设置，请运行 klist -f 命令。0 字符表示禁用了 OK_AS_DELEGATE 标志：

```
$ klist -f
Ticket cache: KEYRING:persistent:0:0
Default principal: admin@EXAMPLE.COM

Valid starting  Expires   Service principal
02/19/2024 09:59:02 02/20/2024 08:21:33 demo/ipa/example.com@EXAMPLE.COM
    Flags: FATO
```

# 第 50 章 使用 PAC 信息加强 KERBEROS 安全性

从 RHEL 8.5 开始，您可以使用默认带有特权属性证书(PAC)信息的身份管理(IdM)。另外，您可以在 RHEL 8.5 之前安装的 IdM 部署中启用安全标识符(SID)。

## 50.1. IDM 中使用的权限属性证书(PAC)

为提高安全性，RHEL Identity Management(IdM)现在在新部署中默认使用 Privilege Attribute 证书(PAC)信息发出 Kerberos 票据。PAC 包含有关 Kerberos 主体的丰富信息，包括其安全标识符(SID)、组成员资格和主目录信息。

默认使用 Microsoft Active Directory(AD)的 SID 是不会重复使用的全局唯一标识符。SIDs express 多个命名空间：每个域都有一个 SID，它是每个对象的 SID 的一个前缀。

从 RHEL 8.5 开始，当安装 IdM 服务器或副本时，安装脚本默认为用户和组群生成 SID。这允许 IdM 使用 PAC 数据。如果您在 RHEL 8.5 前安装 IdM，且您还没有为 AD 域配置信任，您可能没有为 IdM 对象生成 SID。有关为您的 IdM 对象生成 SID 的更多信息，请参阅 IdM 中启用安全标识符(SID)。

通过评估 Kerberos 票据中的 PAC 信息，您可以使用更详细的信息控制资源访问。例如，一个域中的 Administrator 帐户与任何其他域中的 Administrator 帐户唯一不同的 SID。在信任到 AD 域的 IdM 环境中，您可以根据全局唯一 SID 而不是可能在不同位置重复的简单用户名或 UID 来设置访问控制，如 UID 为 0 的每个 Linux root 帐户。

## 50.2. 在 IDM 中启用安全标识符(SID)

如果您在 RHEL 8.5 前安装 IdM，且还没有为 AD 域配置信任，您可能没有为 IdM 对象生成 Security Identifier(SID)。这是因为，在生成 SID 前，生成 SID 的唯一方法是运行 ipa-adtrust-install 命令，将 Trust Controller 角色添加到 IdM 服务器。

从 RHEL 8.6 开始，IdM 中的 Kerberos 要求您的 IdM 对象具有 SID，这是根据权限访问证书(PAC)信息的安全性所必需的。

**先决条件**

- 您在 RHEL 8.5 前安装 IdM。

- 您已运行 ipa-sidgen 任务，它是使用 Active Directory 域配置信任的一部分。
- 您可以作为 IdM admin 帐户进行身份验证。

**流程**

- 启用 SID 使用并触发 SIDgen 任务，以便为现有用户和组生成 SID。此任务可能是资源密集型：

```
[root@server ~]# ipa config-mod --enable-sid --add-sids
```

验证

- 验证 IdM admin 用户帐户条目是否具有 ipantsecurityidentifier 属性，其中 SID 以 -500 结尾，为域管理员保留的 SID:

```
[root@server ~]# ipa user-show admin --all | grep ipantsecurityidentifier
  ipantsecurityidentifier: S-1-5-21-2633809701-976279387-419745629-500
```

**其它资源**

- IdM 中使用的权限属性证书(PAC)
- 如何解决用户无法使用 PAC 认证到 IPA/IDM 的问题 - S4U2PROXY_EVIDENCE_TKT_WITHOUT_PAC 错误 KCS 解决方案
- 信任控制器和信任代理
- 将 SID 配置整合到基本 IPA 安装程序中

# 第 51 章 管理 KERBEROS 票据策略

身份管理(IdM)中的 Kerberos 票据策略对 Kerberos 票据访问、持续时间和续订设置了限制。您可以为运行在 IdM 服务器上的密钥分发中心(KDC)配置 Kerberos 票据策略。

管理 Kerberos 票据策略时会执行以下概念和操作：

- IdM KDC 的角色
- IdM Kerberos 票据策略类型
- Kerberos 认证指示符
- 为 IdM 服务强制执行身份验证指标
- 配置全局票据生命周期策略
- 根据身份验证指标配置全局票据策略
- 为用户配置默认的票据策略
- 为用户配置单独的身份验证指标票据策略
- krbtpolicy-mod 命令的身份验证指标选项

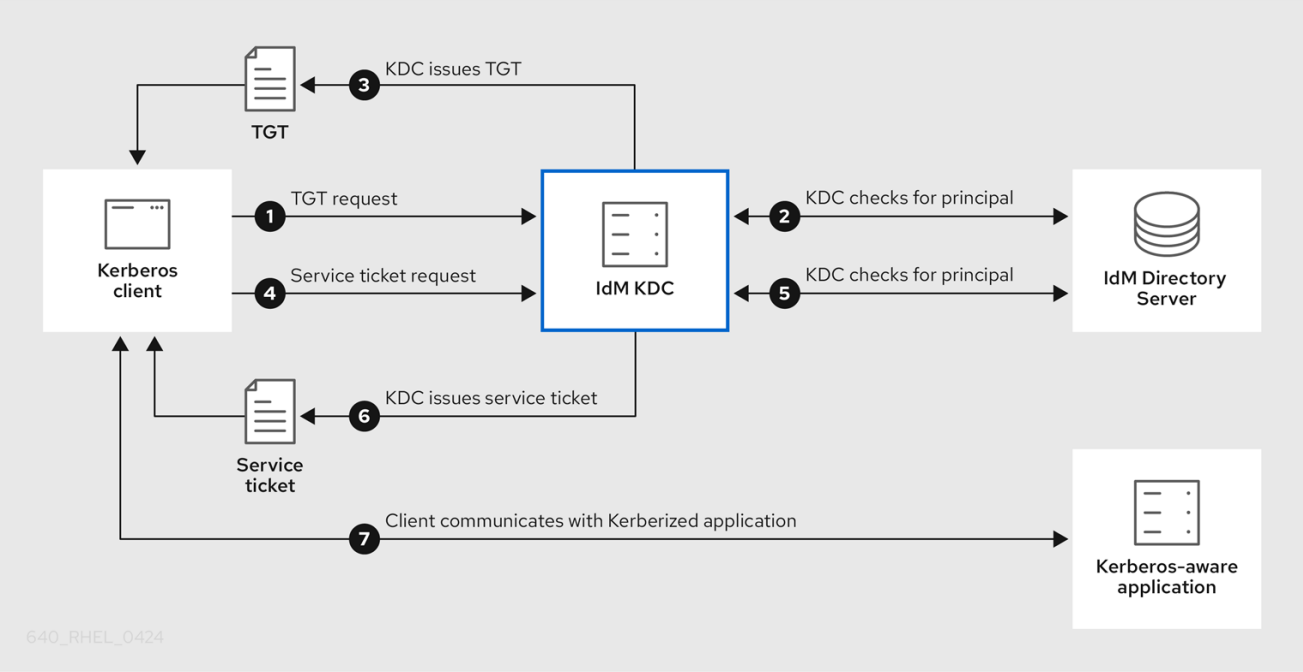
## 51.1. IDM KDC 的角色

身份管理的身份验证机制使用由密钥分发中心(KDC)建立的 Kerberos 基础设施。KDC 是可信赖的权威，其存储凭证信息，并确保来自 IdM 网络内实体的数据的真实性。

每个 IdM 用户、服务和主机都充当 Kerberos 客户端，由唯一的 Kerberos *主体* 识别：

- 对于用户：identifier@REALM，如 admin@EXAMPLE.COM
- 对于服务：service/fully-qualified-hostname@REALM，如 http/server.example.com@EXAMPLE.COM
- 对于主机： host/fully-qualified-hostname@REALM，如 host/client.example.com@EXAMPLE.COM

下图是 Kerberos 客户端、KDC 以及客户端希望与之通信的 Kerberos 应用之间通信的简化。




1. Kerberos 客户端通过作为 Kerberos 主体进行身份验证来向 KDC 识别自己。例如，IdM 用户执行 kinit *username*，并提供其密码。
2. KDC 会检查数据库中的主体，验证客户端，并评估 Kerberos 票据策略 来确定是否授予请求。
3. KDC 根据适当的票据策略，签发一个具有生命周期和 验证指标 的客户端票据授予票(TGT)。

4. 使用 TGT 时，客户端从 KDC 请求 *服务票据*，以便与目标主机上的 Kerberos 服务通信。

5. KDC 检查客户端的 TGT 是否仍然有效，并根据票据策略评估服务票据请求。

6. KDC 向客户端发出 *服务票据*。

7. 通过服务票据，客户端可以在目标主机上启动与服务的 加密通信。

## 51.2. IDM KERBEROS 票据策略类型

IdM Kerberos 票据策略实现以下票据策略类型：

### 连接策略

要保护具有不同安全级别的 Kerberos 服务，您可以定义连接策略来强制执行规则，客户端基于这些规则来检索票据授予票(TGT)。

例如，您可以要求智能卡验证来连接到 client1.example.com，并且需要双因素身份验证来访问 client2.example.com 上的 testservice 应用。

要强制执行连接策略，请将 *身份验证指标* 与服务相关联。只有在服务票据请求中有所需的验证指标的客户端才能访问这些服务。如需更多信息，请参阅 Kerberos 身份验证指标。

### 票据生命周期策略

每个 Kerberos 票据都有一个 *生命周期* 和一个潜在的 *续订期限* ：您可以在达到最长生命周期前续订票据，但不能在超过其最长续订期限之后续订票据。

默认的全局票据生命周期为一天（86400 秒），默认的全局最长续订期限为 1 周（604800 秒）。要调整这些全局值，请参阅 配置全局票据生命周期策略。

您还可以自行定义您自己的票据生命周期策略：

- 要为每个身份验证指标配置不同的全局票据生命周期值，请参阅 根据身份验证指标配置全局票据策略。

- 要为应用任何身份验证方法的单个用户定义票据生命周期值，请参阅 为用户配置默认的票据策略。

- 要为每个只应用到单独用户的身份验证指标定义单个票据生命周期值，请参阅 为用户配置单独的身份验证指标票据策略。

## 51.3. KERBEROS 认证指示符

Kerberos 密钥分发中心(KDC)根据客户端使用哪个预身份验证机制来证明其身份，来将 *身份验证指标* 附加到票据授予票(TGT)：

**otp**

双因素身份验证（密码 + 一次性密码）

**radius**

RADIUS 身份验证（通常用于 802.1x 身份验证）

**pkinit**

PKINIT、智能卡或证书验证

**hardened**

强化的密码（SPAKE 或 FAST）[1]

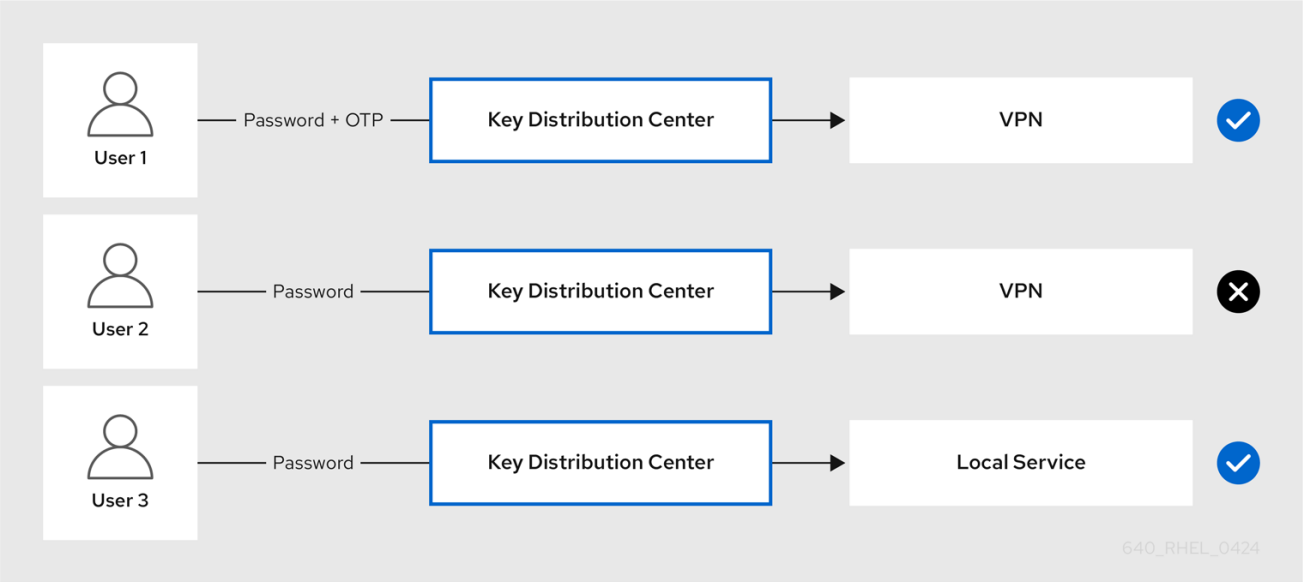
然后 KDC 将来自 TGT 的身份验证指标附加到来自它的任何服务票据请求。KDC 强制执行基于验证指标的策略，如服务访问控制、最长票据生命周期和最长续订期限。

### 身份验证指标和 IdM 服务

如果您将服务或主机与身份验证指标相关联，则只有使用相应身份验证机制获取 TGT 的客户端才能访问它。KDC （不是应用程序或服务），检查服务票证请求中的身份验证指标，并根据 Kerberos 连接策略授予或拒绝请求。

例如，需要双因素身份验证来连接虚拟专用网络(VPN)，请将 otp 身份验证指示与该服务相关联。只有使用一次性密码从 KDC 获得其初始 TGT 的用户才能登录到 VPN：

图 51.1. 需要 otp 验证指示器的 VPN 服务示例




如果服务或主机没有给其分配的身份验证指标，它将接受任何机制验证的票据。

其它资源

- 为 IdM 服务强制执行身份验证指标
- 在 IdM 客户端上为 sudo 启用 GSSAPI 身份验证，并强制实施 Kerberos 身份验证指标

## 51.4. 为 IDM 服务强制执行身份验证指标

身份验证(IdM)支持的验证机制在身份验证强度方面存在差异。例如，使用一次性密码(OTP)与标准密码(OTP)的结合来获取初始 Kerberos 票据授予票(TGT)被视为比仅使用标准密码进行身份验证更加安全。

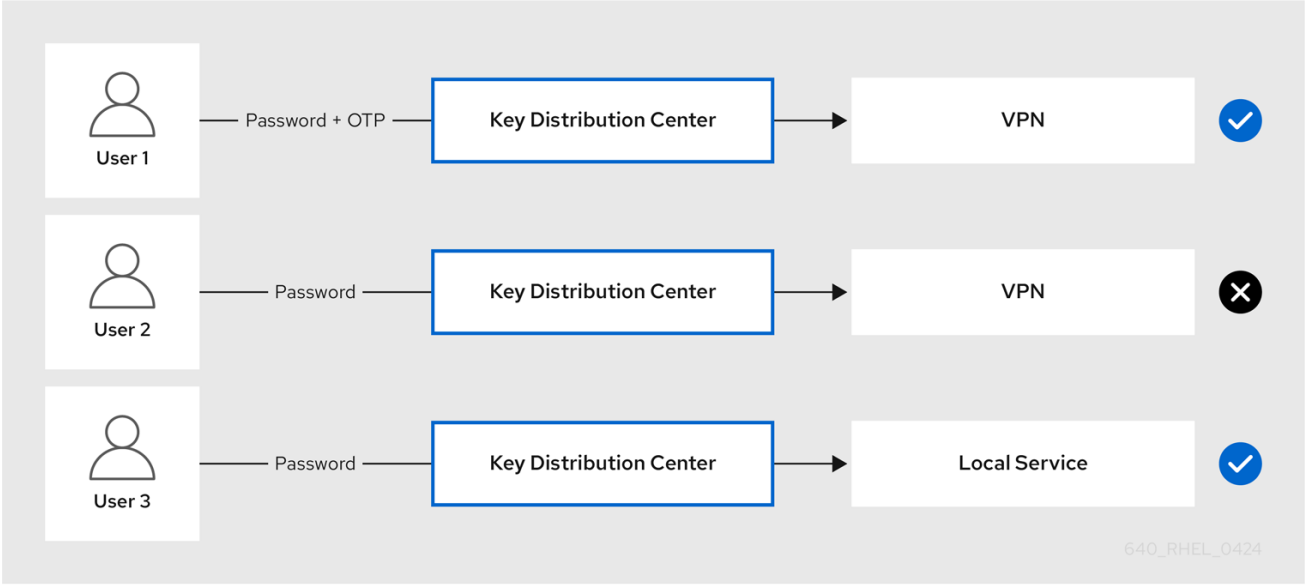
通过将身份验证指示符与特定的 IdM 服务相关联，作为 IdM 管理员，您可以配置服务，以便只有使用这些特定预身份验证机制的用户才能获得他们可以访问该服务的初始 Kerberos 票据授予票(TGT)。

这样，您可以配置不同的 IdM 服务以便：

- 只有使用更强大的身份验证方法获取其初始 TGT（如一次性密码(OTP)）的用户才能访问对安全性至关重要的服务，比如 VPN。

- 使用更简单的身份验证方法获取其初始 TGT（如密码）的用户只能访问非关键服务，如本地登录。

图 51.2. 使用不同技术进行身份验证的示例




这个流程描述了创建 IdM 服务，并将其配置为需要传入的服务票据请求中的特定 Kerberos 身份验证指标。

### 51.4.1. 创建 IdM 服务条目及其 Kerberos keytab

为运行在 IdM 主机上的服务添加 *IdM 服务* 条目会创建相应的 Kerberos 主体，并允许服务请求 SSL 证书、Kerberos keytab 或两者。

以下流程描述了创建 IdM 服务条目，并为加密与该服务的通信生成关联的 Kerberos keytab。

**先决条件**

- 您的服务可以存储 Kerberos 主体、SSL 证书，或两者。

**流程**

1. 使用 ipa service-add 命令添加 IdM 服务，来创建与其关联的 Kerberos 主体。例如，要为运行在主机 client.example.com 上的 testservice 应用程序创建 IdM 服务条目：

```
[root@client ~]# ipa service-add testservice/client.example.com
-------------------------------------------------------------
Modified service "testservice/client.example.com@EXAMPLE.COM"
-------------------------------------------------------------
  Principal name: testservice/client.example.com@EXAMPLE.COM
  Principal alias: testservice/client.example.com@EXAMPLE.COM
  Managed by: client.example.com
```

2. 为客户端上的服务生成并存储 Kerberos keytab。

```
[root@client ~]# ipa-getkeytab -k /etc/testservice.keytab -p
testservice/client.example.com
Keytab successfully retrieved and stored in: /etc/testservice.keytab
```

验证步骤

1. 使用 ipa service-show 命令显示 IdM 服务的信息。

```
[root@server ~]# ipa service-show testservice/client.example.com
  Principal name: testservice/client.example.com@EXAMPLE.COM
  Principal alias: testservice/client.example.com@EXAMPLE.COM
  Keytab: True
  Managed by: client.example.com
```

2. 使用 klist 命令显示服务的 Kerberos keytab 的内容。

```
[root@server etc]# klist -ekt /etc/testservice.keytab
Keytab name: FILE:/etc/testservice.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   2 04/01/2020 17:52:55 testservice/client.example.com@EXAMPLE.COM (aes256-cts-
hmac-sha1-96)
   2 04/01/2020 17:52:55 testservice/client.example.com@EXAMPLE.COM (aes128-cts-
hmac-sha1-96)
   2 04/01/2020 17:52:55 testservice/client.example.com@EXAMPLE.COM (camellia128-
cts-cmac)
   2 04/01/2020 17:52:55 testservice/client.example.com@EXAMPLE.COM (camellia256-
cts-cmac)
```

### 51.4.2. 使用 IdM CLI 将身份验证指示符与 IdM 服务相关联

作为身份管理(IdM)管理员，您可以配置主机或服务，来要求客户端应用程序提供的服务票据包含特定的验证指标。例如，您可以确保在获取 Kerberos 票据授予票据(TGT)时，只有使用有效的带有密码的 IdM 双因素身份验证令牌的用户才能访问该主机或服务。

按照以下流程配置服务，以要求来自传入服务票据请求的特定 Kerberos 身份验证指标。

**先决条件**

- 您已为运行在 IdM 主机上的服务创建了 IdM 服务条目。请参阅 创建 IdM 服务条目及其 Kerberos keytab。
- 您已在 IdM 中获得了管理用户的票据授予票据。

**警告**

不要 将身份验证指标分配给内部 IdM 服务。以下 IdM 服务无法执行 PKINIT 和多因素身份验证方法所需的交互式身份验证步骤：

```
host/server.example.com@EXAMPLE.COM
HTTP/server.example.com@EXAMPLE.COM
ldap/server.example.com@EXAMPLE.COM
DNS/server.example.com@EXAMPLE.COM
cifs/server.example.com@EXAMPLE.COM
```

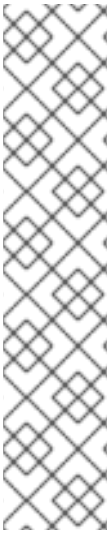
**流程**

- 使用 ipa service-mod 命令为服务指定一个或多个所需的身份验证指标，用 --auth-ind 参数标识。

| 身份验证方法 | --auth-ind 值 |
| --- | --- |
| 双因素身份验证 | **otp** |
| RADIUS 身份验证 | **radius** |
| PKINIT、智能卡或证书验证 | **pkinit** |
| 强化的密码（SPAKE 或 FAST） | **hardened** |

例如，要求用户通过智能卡或 OTP 身份验证来检索主机 client.example.com 上 testservice 主体的服务票据：

```
[root@server ~]# ipa service-mod testservice/client.example.com@EXAMPLE.COM --
auth-ind otp --auth-ind pkinit
-----------------------------------------------------------
Modified service "testservice/client.example.com@EXAMPLE.COM"
-----------------------------------------------------------
  Principal name: testservice/client.example.com@EXAMPLE.COM
  Principal alias: testservice/client.example.com@EXAMPLE.COM
  Authentication Indicators: otp, pkinit
  Managed by: client.example.com
```



**注意**

要从服务中删除所有验证指标，请提供一个空的指标列表：

```
[root@server ~]# ipa service-mod
testservice/client.example.com@EXAMPLE.COM --auth-ind ''
-----------------------------------------------------
Modified service "testservice/client.example.com@EXAMPLE.COM"
-----------------------------------------------------
  Principal name: testservice/client.example.com@EXAMPLE.COM
  Principal alias: testservice/client.example.com@EXAMPLE.COM
  Managed by: client.example.com
```

**验证步骤**

- 使用 ipa service-show 命令显示关于 IdM 服务的信息，包括其所需的身份验证指标。

```
[root@server ~]# ipa service-show testservice/client.example.com
  Principal name: testservice/client.example.com@EXAMPLE.COM
  Principal alias: testservice/client.example.com@EXAMPLE.COM
  Authentication Indicators: otp, pkinit
  Keytab: True
  Managed by: client.example.com
```

**其它资源**

- 为 IdM 服务检索 Kerberos 服务票据
- 在 IdM 客户端上为 sudo 启用 GSSAPI 身份验证，并强制实施 Kerberos 身份验证指标

### 51.4.3. 使用 IdM Web UI 将验证指标与 IdM 服务关联

作为身份管理(IdM)管理员，您可以配置主机或服务，以便客户端应用程序所提供的服务票据包含特定的身份验证指标。例如，您可以确保在获取 Kerberos 票据授予票据(TGT)时，只有使用带有密码的有效的 IdM 双因素身份验证令牌的用户才能访问该主机或服务。

按照以下流程，使用 IdM Web UI 配置主机或服务，以要求来自传入票据请求的特定的 Kerberos 身份验证指标。

**先决条件**

- 您以管理用户的身份已登录到 IdM Web UI。

**流程**

1. 选择 **Identity → Hosts** 或 **Identity → Services**。
2. 单击所需的主机或服务的名称。
3. 在 **Authentication indicators** 下，选择所需的验证方法。
   - 例如，选择 **OTP** 来确保在获取 Kerberos TGT 时，只有使用带有密码的有效的 IdM 双因素身份验证令牌的用户才能访问主机或服务。
   - 如果您选择 **OTP** 和 **RADIUS**，那么在获取 Kerberos TGT 时使用带有密码的有效的 IdM 双因素身份验证令牌的用户,以及 使用 RADIUS 服务器获取 Kerberos TGT 的用户，都将被允许访问。
4. 点击页面顶部的 **Save**。

**其它资源**

- 为 IdM 服务检索 Kerberos 服务票据
- 在 IdM 客户端上为 sudo 启用 GSSAPI 身份验证，并强制实施 Kerberos 身份验证指标

### 51.4.4. 为 IdM 服务检索 Kerberos 服务票据

以下流程描述了为 IdM 服务检索 Kerberos 服务票据。您可以使用此流程来测试 Kerberos 票据策略，比如强制票据授予票据(TGT)中存在某些 Kerberos 验证指标。

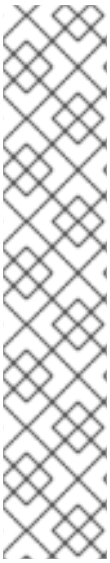
**先决条件**

- 如果您正在使用的服务不是内部 IdM 服务，您已为其创建了相应的 *IdM 服务* 条目。请参阅 创建 IdM 服务条目及其 Kerberos keytab。
- 您有一个 Kerberos 票据授予票据(TGT)。

**流程**

- 使用带-S 选项的 kvno 命令来检索服务票据，并指定 IdM 服务的名称和管理它的主机的完全限定域名。

```
[root@server ~]# kvno -S testservice client.example.com
testservice/client.example.com@EXAMPLE.COM: kvno = 1
```



**注意**

如果您需要访问 IdM 服务以及当前的票据授予票据(TGT)没有所需的与之关联的 Kerberos 身份验证指标，请使用 kdestroy 命令清除当前的 Kerberos 凭证缓存，并检索新的 TGT：

```
[root@server ~]# kdestroy
```

例如，如果您最初通过使用密码的身份验证来获取了 TGT，并且您需要访问具有与之相关联的 pkinit身份验证指标的 IdM 服务，请销毁当前的凭证缓存，并使用智能卡重新进行身份验证。请参阅 Kerberos 身份验证指标。

**验证步骤**

- 使用 klist 命令来验证服务票据是否在默认的 Kerberos 凭据缓存中。

```
[root@server etc]# klist_
Ticket cache: KCM:1000
Default principal: admin@EXAMPLE.COM

Valid starting     Expires            Service principal
```

```
04/01/2020 12:52:42  04/02/2020 12:52:39  krbtgt/EXAMPLE.COM@EXAMPLE.COM
04/01/2020 12:54:07 04/02/2020 12:52:39
testservice/client.example.com@EXAMPLE.COM
```

### 51.4.5. 其它资源

- 请参阅 Kerberos 身份验证指标。

## 51.5. 配置全局票据生命周期策略

全局票据策略适用于所有服务票据，也适用于没有定义任何按用户的票据策略的用户。

以下流程描述了使用 ipa krbtpolicy-mod 命令调整全局 Kerberos 票据策略的最大票据生命周期和最大票据续订期限。

使用 ipa krbtpolicy-mod 命令时，至少指定以下参数之一：

- --maxlife 最长票据生命周期（以秒为单位）
- --maxrenew 最长续订期限（以秒为单位）

**流程**

1. 修改全局票据策略：

```
[root@server ~]# ipa krbtpolicy-mod --maxlife=$((8*60*60)) --maxrenew=$((24*60*60))
  Max life: 28800
  Max renew: 86400
```

在本例中，最长生命周期设置为 8 小时（8 * 60 分钟 * 60 秒），最长续订期限设置为一天（24 * 60 分钟 * 60 秒）。

2. 可选：将全局 Kerberos 票据策略重置为默认安装值：

```
[root@server ~]# ipa krbtpolicy-reset
  Max life: 86400
  Max renew: 604800
```

验证步骤

- 显示全局票据策略：

```
[root@server ~]# ipa krbtpolicy-show
  Max life: 28800
  Max renew: 86640
```

其它资源

- 请参阅 为用户配置默认的票据策略。
- 请参阅 为用户配置单个的身份验证指标票据策略。

### 51.6. 根据身份验证指标配置全局票据策略

按照以下流程，为每个身份验证指标调整全局最长票据生命周期和最长可续订期限。这些设置适用于没有定义按用户的票据策略的用户。

使用 ipa krbtpolicy-mod 命令来指定 Kerberos 票据的全局最长生命周期或最大可用期限，具体取决于它们所附加的 身份验证指标。

流程

- 例如，将全局双因素票据生命周期和续订期限值设置为一周，将全局智能卡票据生命周期和续订期限值设置为两周：

```
[root@server ~]# ipa krbtpolicy-mod --otp-maxlife=604800 --otp-maxrenew=604800 --
pkinit-maxlife=172800 --pkinit-maxrenew=172800
```

验证步骤

- 显示全局票据策略：

```
[root@server ~]# ipa krbtpolicy-show
```

```
Max life: 86400
OTP max life: 604800
PKINIT max life: 172800
Max renew: 604800
OTP max renew: 604800
PKINIT max renew: 172800
```

请注意，OTP 和 PKINIT 值与全局默认的 Max life 和 Max renew 值不同。

**其它资源**

- 请参阅 krbtpolicy-mod 命令的身份验证指标选项。
- 请参阅 为用户配置默认的票据策略。
- 请参阅 为用户配置单个的身份验证指标票据策略。

## 51.7. 为用户配置默认的票据策略

您可以定义一个仅适用于单个用户的 Kerberos 票据策略。这些按用户的设置会覆盖所有验证指标的全局票据策略。

使用 ipa krbtpolicy-mod *username* 命令，并至少指定以下参数之一：

- **--maxlife** 最长票据生命周期（以秒为单位）
- **--maxrenew** 最长续订期限（以秒为单位）

**流程**

1. 例如，将 IdM admin 用户的最长票据生命周期设置为两天，将最长续订期限设置为 2 周：

```
[root@server ~]# ipa krbtpolicy-mod admin --maxlife=172800 --maxrenew=1209600
  Max life: 172800
  Max renew: 1209600
```

2. 可选：为用户重置票据策略：

```
[root@server ~]# ipa krbtpolicy-reset admin
```

验证步骤

- 显示应用到用户的有效 Kerberos 票据策略：

```
[root@server ~]# ipa krbtpolicy-show admin
  Max life: 172800
  Max renew: 1209600
```

其它资源

- 请参阅 配置全局票据生命周期策略。
- 请参阅 配置每个验证指标的全局票据策略。

## 51.8. 为用户配置单独的身份验证指标票据策略

作为管理员，您可以为每个身份验证指标不同的用户定义 Kerberos 票据策略。例如，您可以将策略配置为允许 IdM admin 用户续订两天的票据（如果是通过 OTP 身份验证获取的票据）；或者续订一周的票据（是通过智能卡身份验证获取的票据）。

这些按身份验证的指标设置将覆盖 *用户的* 默认票据策略、*全局的* 默认票据策略，以及任何 *全局的* 身份验证指标票据策略。

使用 ipa krbtpolicy-mod *username* 命令，为用户的 Kerberos 票据设置自定义的最长生命周期和最长可续订期限值，具体取决于附加给它们的 身份验证指标 。

流程

1. 例如，要允许 IdM admin 用户续订两天的 Kerberos 票据（如果是使用一次性密码身份验证获取的），请设置 --otp-maxrenew 选项：

```
[root@server ~]# ipa krbtpolicy-mod admin --otp-maxrenew=$((2*24*60*60))
  OTP max renew: 172800
```

2. 可选：为用户重置票据策略：

```
[root@server ~]# ipa krbtpolicy-reset username
```

验证步骤

- 显示应用到用户的有效 Kerberos 票据策略：

```
[root@server ~]# ipa krbtpolicy-show admin
  Max life: 28800
  Max renew: 86640
```

其它资源

- 请参阅 krbtpolicy-mod 命令的身份验证指标选项。
- 请参阅 为用户配置默认的票据策略。
- 请参阅 配置全局票据生命周期策略。
- 请参阅 配置每个验证指标的全局票据策略。

## 51.9. KRBTPOLICY-MOD 命令的身份验证指标选项

使用以下参数为身份验证指标指定值：

表 51.1. krbtpolicy-mod 命令的身份验证指标选项

| 身份验证指标 | 最长生命周期的参数 | 最长续订期限的参数 |
| --- | --- | --- |
| otp | --otp-maxlife | --otp-maxrenew |
| radius | --radius-maxlife | --radius-maxrenew |
| pkinit | --pkinit-maxlife | --pkinit-maxrenew |
| hardened | --hardened-maxlife | --hardened-maxrenew |

[1] 通过使用单方公钥认证的密钥交换(SPAKE)预认证和/或通过安全隧道(FAST)保护的验证，可保护强化的密码免于暴力密码字典攻击。

# 第 52 章 IDM 中的 KERBEROS PKINIT 身份验证

Kerberos (PKINIT)中初始身份验证的公钥加密是 Kerberos 的预身份验证机制。身份管理(IdM)服务器包含一个用于 Kerberos PKINIT 身份验证的机制。

## 52.1. 默认 PKINIT 配置

IdM 服务器上的默认 PKINIT 配置取决于证书颁发机构(CA)配置。

**表 52.1. IdM 中的默认 PKINIT 配置**

| CA 配置 | PKINIT 配置 |
| --- | --- |
| 没有 CA，没有提供外部 PKINIT 证书 | 本地 PKINIT：IdM 仅将 PKINIT 用于服务器上的内部目的。 |
| 没有 CA，向 IdM 提供外部 PKINIT 证书 | IdM 使用外部 Kerberos 密钥分发中心(KDC)证书和 CA 证书来配置 PKINIT。 |
| 带有集成的 CA | IdM 使用 IdM CA 签名的证书配置 PKINIT。 |

## 52.2. 显示当前 PKINIT 配置

IdM 提供多个命令，您可用来查询域中的 PKINIT 配置。

**流程**

- 要确定域中的 PKINIT 状态，请使用 ipa pkinit-status 命令：

```
$ ipa pkinit-status
  Server name: server1.example.com
  PKINIT status: enabled
  [...output truncated...]
  Server name: server2.example.com
  PKINIT status: disabled
  [...output truncated...]
```

命令将 PKINIT 配置状态显示为 enabled 或 disabled：

  - enabled: PKINIT 是使用集成 IdM CA 或外部 PKINIT 证书签名的证书配置的。

- disabled ：IdM 仅将 PKINIT 用于 IdM 服务器上的内部目的。

- 要列出支持 IdM 客户端 PKINIT 的活跃的 Kerberos 密钥分发中心(KDC)的 IdM 服务器，请在任何服务器上使用 ipa config-show 命令：

```
$ ipa config-show
  Maximum username length: 32
  Home directory base: /home
  Default shell: /bin/sh
  Default users group: ipausers
  [...output truncated...]
  IPA masters capable of PKINIT: server1.example.com
  [...output truncated...]
```

### 52.3. 在 IDM 中配置 PKINIT

如果您的 IdM 服务器在 PKINIT 被禁用的情况下运行，请使用这些步骤启用它。例如，如果您传递了 ipa-server-install 或 ipa-replica-install 工具和 --no-pkinit 选项，则服务器在禁用了 PKINIT 的情况下运行。

**先决条件**

- 确保安装了证书颁发机构(CA)的所有 IdM 服务器都在同一域级别上运行。

**流程**

1. 检查服务器上是否启用了 PKINIT：

```
# kinit admin

Password for admin@IDM.EXAMPLE.COM:
# ipa pkinit-status --server=server.idm.example.com
1 server matched
----------------
Server name: server.idm.example.com
PKINIT status:enabled
----------------------------
Number of entries returned 1
----------------------------
```

如果 PKINIT 被禁用了，您将看到以下输出：

```
# ipa pkinit-status --server server.idm.example.com
----------------
0 servers matched
----------------
---------------------------
Number of entries returned 0
---------------------------
```

如果省略了 **--server <server_fqdn>** 参数，您也可以使用命令来查找所有启用了 **PKINIT** 的服务器。

2. 如果您使用没有 **CA** 的 **IdM**：

   a. 在 **IdM** 服务器上，安装签名了 **Kerberos** 密钥分发中心**(KDC)**证书的 **CA** 证书：

   ```
   # ipa-cacert-manage install -t CT,C,C ca.pem
   ```

   b. 要更新所有 **IPA** 主机，请在所有副本和客户端上重复 **ipa-certupdate** 命令：

   ```
   # ipa-certupdate
   ```

   c. 使用 **ipa-cacert-manage list** 命令检查 **CA** 证书是否已添加。例如：

   ```
   # ipa-cacert-manage list
   CN=CA,O=Example Organization
   The ipa-cacert-manage command was successful
   ```

   d. 使用 **ipa-server-certinstall** 工具安装外部 **KDC** 证书。**KDC** 证书必须满足以下条件：

   - 它使用通用名称 **CN=*fully_qualified_domain_name,certificate_subject_base*** 发布。
   - 它包括 **Kerberos** 主体 **krbtgt/*REALM_NAME@REALM_NAME***。
   - 它包含 **KDC** 身份验证的对象标识符**(OID)**：**1.3.6.1.5.2.3.5**。

```
# ipa-server-certinstall --kdc kdc.pem kdc.key

# systemctl restart krb5kdc.service
```

e. 查看您的 **PKINIT** 状态：

```
# ipa pkinit-status
  Server name: server1.example.com
  PKINIT status: enabled
  [...output truncated...]
  Server name: server2.example.com
  PKINIT status: disabled
  [...output truncated...]
```

3. 如果您带有 **CA** 证书的 **IdM**，请启用 **PKINIT**，如下所示：

```
# ipa-pkinit-manage enable
  Configuring Kerberos KDC (krb5kdc)
  [1/1]: installing X509 Certificate for PKINIT
  Done configuring Kerberos KDC (krb5kdc).
  The ipa-pkinit-manage command was successful
```

如果您使用 **IdM CA**，则命令会从 **CA** 请求 **PKINIT KDC** 证书。

**其它资源**

- **ipa-server-certinstall(1)** 手册页

### 52.4. 其它资源

- 有关 **Kerberos PKINIT** 的详情，请参阅 **MIT Kerberos** 文档中的 **PKINIT 配置**。

# 第 53 章 维护 IDM KERBEROS KEYTAB 文件

了解更多有关 Kerberos keytab 文件是什么以及身份管理(IdM)如何使用它们来允许服务使用 Kerberos 安全地进行身份验证。

您可以使用这些信息来了解您应该保护这些敏感文件的原因，并对 IdM 服务之间的通信问题进行故障排除。

如需更多信息，请参阅以下主题：

- Identity Management 如何使用 Kerberos keytab 文件
- 验证 Kerberos keytab 文件是否与 IdM 数据库同步
- IdM Kerberos keytab 文件及其内容列表
- 查看 IdM 主密钥的加密类型。

## 53.1. IDENTITY MANAGEMENT 如何使用 KERBEROS KEYTAB 文件

Kerberos keytab 是包含 Kerberos 主体及其对应的加密密钥的文件。主机、服务、用户和脚本可以使用 keytabs 来安全地向 Kerberos 密钥分发中心(KDC)进行身份验证，而无需人为干预。

IdM 服务器中的每个 IdM 服务都有一个存储在 Kerberos 数据库中的唯一 Kerberos 主体。例如，如果 IdM 服务器 east.idm.example.com 和 west.idm.example.com 提供 DNS 服务，IdM 会创建 2 个唯一的 DNS Kerberos 主体来识别这些服务，它遵循命名规则 <service>/host.domain.com@REALM.COM:

- DNS/east.idm.example.com@IDM.EXAMPLE.COM
- DNS/west.idm.example.com@IDM.EXAMPLE.COM

IdM 在服务器上为这些服务的每一个创建一个 keytab，以存储 Kerberos 密钥的本地副本，以及它们

的密钥版本号(KVNO)。例如，默认 keytab 文件 /etc/krb5.keytab 存储 主机 主体，它代表了在 Kerberos 域中的机器，用于登录身份验证。KDC 为它支持的不同加密算法生成加密密钥，如 aes256-cts-hmac-sha1-96 和 aes128-cts-hmac-sha1-96。

您可以使用 klist 命令显示 keytab 文件的内容：

```
[root@idmserver ~]# klist -ekt /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   2 02/24/2022 20:28:09 host/idmserver.idm.example.com@IDM.EXAMPLE.COM (aes256-cts-hmac-sha1-96)
   2 02/24/2022 20:28:09 host/idmserver.idm.example.com@IDM.EXAMPLE.COM (aes128-cts-hmac-sha1-96)
   2 02/24/2022 20:28:09 host/idmserver.idm.example.com@IDM.EXAMPLE.COM (camellia128-cts-cmac)
   2 02/24/2022 20:28:09 host/idmserver.idm.example.com@IDM.EXAMPLE.COM (camellia256-cts-cmac)
```

其它资源

- 验证 Kerberos keytab 文件是否与 IdM 数据库同步
- IdM Kerberos keytab 文件及其内容列表

## 53.2. 验证 KERBEROS KEYTAB 文件是否与 IDM 数据库同步

当您更改 Kerberos 密码时，IdM 会自动生成新的对应的 Kerberos 密钥并增加其密钥版本号(KVNO)。如果没有使用新密钥和 KVNO 更新 Kerberos keytab，则任何依赖于 keytab 获取有效密钥的服务可能无法验证 Kerberos 密钥分发中心(KDC)。

如果您的 IdM 服务无法与其他服务通信，请使用以下步骤验证您的 Kerberos keytab 文件与存储在 IdM 数据库中的密钥同步。如果它们没有同步，请使用更新的密钥和 KVNO 来检索 Kerberos keytab。这个示例比较，并检索 IdM 服务器更新的 DNS 主体。

先决条件

- 您必须作为 IdM admin 帐户进行身份验证才能检索 keytab 文件
- 您必须以 root 帐户的身份进行身份验证，才能修改其他用户所拥有的 keytab 文件

**流程**

1. 在您要验证的 keytab 中显示主体的 KVNO。在以下示例中，/etc/named.keytab 文件具有 KVNO 为 2 的 DNS/server1.idm.example.com@EXAMPLE.COM 主体的密钥。

```
[root@server1 ~]# klist -ekt /etc/named.keytab
Keytab name: FILE:/etc/named.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   2 11/26/2021 13:51:11 DNS/server1.idm.example.com@EXAMPLE.COM (aes256-cts-
hmac-sha1-96)
   2 11/26/2021 13:51:11 DNS/server1.idm.example.com@EXAMPLE.COM (aes128-cts-
hmac-sha1-96)
   2 11/26/2021 13:51:11 DNS/server1.idm.example.com@EXAMPLE.COM (camellia128-
cts-cmac)
   2 11/26/2021 13:51:11 DNS/server1.idm.example.com@EXAMPLE.COM (camellia256-
cts-cmac)
```

2. 显示存储在 IdM 数据库中的主体的 KVNO。在这个示例中，IdM 数据库中密钥的 KVNO 与 keytab 中的 KVNO 不匹配。

```
[root@server1 ~]# kvno DNS/server1.idm.example.com@EXAMPLE.COM
DNS/server1.idm.example.com@EXAMPLE.COM: kvno = 3
```

3. 作为 IdM admin 帐户进行身份验证。

```
[root@server1 ~]# kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

4. 为主体检索更新的 Kerberos 密钥并将其存储在其 keytab 中。以 root 用户身份执行此步骤，以便您可以修改 /etc/named.keytab 文件，该文件归 named 用户所有。

```
[root@server1 ~]# ipa-getkeytab -s server1.idm.example.com -p
DNS/server1.idm.example.com -k /etc/named.keytab
```

验证

1. 在 keytab 中显示主体的更新 KVNO。

```
[root@server1 ~]# klist -ekt /etc/named.keytab
Keytab name: FILE:/etc/named.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   4 08/17/2022 14:42:11 DNS/server1.idm.example.com@EXAMPLE.COM (aes256-cts-
```

```
hmac-sha1-96)
   4 08/17/2022 14:42:11 DNS/server1.idm.example.com@EXAMPLE.COM (aes128-cts-
hmac-sha1-96)
   4 08/17/2022 14:42:11 DNS/server1.idm.example.com@EXAMPLE.COM (camellia128-
cts-cmac)
   4 08/17/2022 14:42:11 DNS/server1.idm.example.com@EXAMPLE.COM (camellia256-
cts-cmac)
```

2. 显示 IdM 数据库中存储的主体的 KVNO，并确保它与 keytab 中的 KVNO 匹配。

```
[root@server1 ~]# kvno DNS/server1.idm.example.com@EXAMPLE.COM
DNS/server1.idm.example.com@EXAMPLE.COM: kvno = 4
```

**其它资源**

- Identity Management 如何使用 Kerberos keytab 文件
- IdM Kerberos keytab 文件及其内容列表

### 53.3. IDM KERBEROS KEYTAB 文件及其内容列表

下表显示了 IdM Kerberos keytab 文件的位置、内容和目的。

**表 53.1. 表**

| keytab 位置 | 内容 | 目的 |
|---|---|---|
| **/etc/krb5.keytab** | **主机** 主体 | 如果没有 **nfs** 主体，在登录时验证用户凭证供 NFS 使用 |
| **/etc/dirsrv/ds.keytab** | **LDAP** 主体 | 对 IdM 数据库进行身份验证，安全地在 IdM 副本之间复制数据库内容 |
| **/var/lib/ipa/gssproxy/http.keytab** | **HTTP** 主体 | 向 Apache 服务器进行身份验证 |
| **/etc/named.keytab** | **DNS** 主体 | 安全更新 DNS 记录 |
| **/etc/ipa/dnssec/ipa-dnskeysyncd.keytab** | **ipa-dnskeysyncd** principal | 使 OpenDNSSEC 与 LDAP 同步 |
| **/etc/pki/pki-tomcat/dogtag.keytab** | **dogtag** principal | 与证书颁发机构(CA)通信. |

| keytab 位置 | 内容 | 目的 |
| --- | --- | --- |
| **/etc/samba/samba.keytab** | **CIFS** 和 **主机** 主体 | 与 Samba 服务通信 |
| **/var/lib/sss/keytabs/*ad-domain.com*.keytab** | Active Directory(AD)域控制器(DCs)主体以 ***HOSTNAME$@AD-DOMAIN.COM***格式 | 通过 IdM-AD Trust 与 AD DC 进行通讯 |

**其它资源**

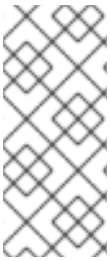
- Identity Management 如何使用 Kerberos keytab 文件
- 验证 Kerberos keytab 文件是否与 IdM 数据库同步

## 53.4. 查看 IDM 主密钥的加密类型

作为身份管理(IdM)管理员，您可以查看 IdM 主密钥的加密类型，这是 IdM Kerberos 分发中心(KDC)在静态存储所有其他主体的密钥时用来加密它们的密钥。了解加密类型可帮助您确定部署与 FIPS 标准的兼容性。

从 RHEL 8.7 开始，加密类型是 aes256-cts-hmac-sha384-192。这个加密类型与旨在遵守 FIPS 140-3 的默认的 RHEL 9 FIPS 加密策略兼容。

之前 RHEL 版本中使用的加密类型与遵守 FIPS 140-3 标准的 RHEL 9 系统不兼容。要使 FIPS 模式下的 RHEL 9 系统与 RHEL 8 FIPS 140-2 部署兼容，请在 RHEL 9 系统上启用 FIPS:AD-SUPPORT 加密策略。



**注意**

Microsoft 的活动目录实现尚不支持任何使用 SHA-2 HMAC 的 RFC8009 Kerberos 加密类型。如果您配置了 IdM-AD 信任，即使 IdM 主密钥的加密类型是 aes256-cts-hmac-sha384-192，您也需要使用 FIPS:AD-SUPPORT 加密子策略。

**先决条件**

- 您有访问 IdM 部署中任何 RHEL 8 副本的 root 权限。

**流程**

- 在副本上，在命令行界面上查看加密类型：

```
# kadmin.local getprinc K/M | grep -E '^Key:'
Key: vno 1, aes256-cts-hmac-sha1-96
```

输出中的 **aes256-cts-hmac-sha1-96** 键表示 IdM 部署已安装在运行 RHEL 8.6 或更早版本的服务器上。输出中存在 **aes256-cts-hmac-sha384-192** 键表示 IdM 部署被安装在运行 RHEL 8.7 或更高版本的服务器上。

# 第 54 章 在 IDM 中使用 KDC 代理

有些管理员可能会选择使默认的 Kerberos 端口在部署中无法访问。要允许用户、主机和服务获取 Kerberos 凭据，您可以使用 HTTPS 服务作为代理，其通过 HTTPS 端口 443 与 Kerberos 进行通信的。

在身份管理(IdM)中，Kerberos 密钥分发中心代理 (KKDCP)提供此功能。

在 IdM 服务器上，KKDCP 默认启用，并通过 https://*server.idm.example.com*/KdcProxy 提供。在 IdM 客户端上，您必须更改其 Kerberos 配置来访问 KKDCP。

## 54.1. 配置 IDM 客户端以使用 KKDCP

作为身份管理(IdM)系统管理员，您可以将 IdM 客户端配置为使用 IdM 服务器上的 Kerberos 密钥分发中心代理(KKDCP)。如果默认的 Kerberos 端口在 IdM 服务器上无法访问，并且 HTTPS 端口 443 是访问 Kerberos 服务的唯一方式，那么这很有用。

**先决条件**

- 您有访问 IdM 客户端的 root 权限。

**流程**

1. 打开 /etc/krb5.conf 文件进行编辑。

2. 在 [realms] 部分中，对 kdc、admin_server 和 kpasswd_server 选项输入 KKDCP 的 URL：

```
[realms]
EXAMPLE.COM = {
 kdc = https://kdc.example.com/KdcProxy
 admin_server = https://kdc.example.com/KdcProxy
 kpasswd_server = https://kdc.example.com/KdcProxy
 default_domain = example.com
}
```

要实现冗余，您可以多次添加参数 kdc、admin_server 和 kpasswd_server 来指示不同的 KKDCP 服务器。

3. 重启 sssd 服务以使更改生效：

```
~]# systemctl restart sssd
```

### 54.2. 验证 IDM 服务器上是否启用了 KKDCP

在身份管理(IdM)服务器上，如果属性和值对 ipaConfigString=kdcProxyEnabled 在目录中存在，则每次 Apache Web 服务器启动时，Kerberos 密钥分发中心代理(KKDCP)会自动启用。在这种情况下，将创建符号链接 /etc/httpd/conf.d/ipa-kdc-proxy.conf。

即使作为非特权用户，您也可以验证 IdM 服务器上是否启用了 KKDCP。

**流程**

- 检查符号链接是否存在：

```
$ ls -l /etc/httpd/conf.d/ipa-kdc-proxy.conf
lrwxrwxrwx. 1 root root 36 Jun 21  2020 /etc/httpd/conf.d/ipa-kdc-proxy.conf ->
/etc/ipa/kdcproxy/ipa-kdc-proxy.conf
```

输出确认启用了 KKDCP。

### 54.3. 在 IDM 服务器上禁用 KKDCP

作为身份管理(IdM)系统管理员，您可以在 IdM 服务器上禁用 Kerberos 密钥分发中心代理(KKDCP)。

**先决条件**

- 您有访问 IdM 服务器的 root 权限。

**流程**

1. 从目录中删除 ipaConfigString=kdcProxyEnabled 属性和值对：

```
# ipa-ldap-updater /usr/share/ipa/kdcproxy-disable.uldif
Update complete
The ipa-ldap-updater command was successful
```

2. 重启 httpd 服务：

```
# systemctl restart httpd.service
```

KKDCP 现在在当前的 IdM 服务器上被禁用。

验证步骤

- 验证符号链接不存在：

```
$ ls -l /etc/httpd/conf.d/ipa-kdc-proxy.conf
ls: cannot access '/etc/httpd/conf.d/ipa-kdc-proxy.conf': No such file or directory
```

### 54.4. 在 IDM 服务器上重新启用 KKDCP

在 IdM 服务器上，默认启用 Kerberos 密钥分发中心代理(KKDCP)，并可通过 https://*server.idm.example.com*/KdcProxy 获取。

如果服务器上已禁用了 KKDCP，您可以重新启用它。

先决条件

- 您有访问 IdM 服务器的 root 权限。

流程

1. 将 ipaConfigString=kdcProxyEnabled 属性和值对添加到目录中：

```
# ipa-ldap-updater /usr/share/ipa/kdcproxy-enable.uldif
Update complete
The ipa-ldap-updater command was successful
```

2. 重启 httpd 服务：

```
# systemctl restart httpd.service
```

KKDCP 现在在当前的 IdM 服务器上被启用。

验证步骤

- 验证符号链接是否存在：

```
$ ls -l /etc/httpd/conf.d/ipa-kdc-proxy.conf
lrwxrwxrwx. 1 root root 36 Jun 21  2020 /etc/httpd/conf.d/ipa-kdc-proxy.conf ->
/etc/ipa/kdcproxy/ipa-kdc-proxy.conf
```

### 54.5. 配置 KKDCP 服务器 I

使用以下配置，您可以启用 TCP 作为 IdM KKDCP 和 活动目录(AD)域之间的传输协议，其中会使用多个 Kerberos 服务器。

先决条件

- 您有 root 访问权限。

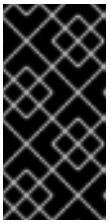
流程

1. 将 /etc/ipa/kdcproxy/kdcproxy.conf 文件的 [global] 部分中的 use_dns 参数设为 false。

```
[global]
use_dns = false
```

2. 将代理域信息放在 /etc/ipa/kdcproxy/kdcproxy.conf 文件中。例如，对于具有代理的 [AD.*EXAMPLE.COM*] 域，请按如下所示列出域配置参数：

```
[AD.EXAMPLE.COM]
kerberos = kerberos+tcp://1.2.3.4:88 kerberos+tcp://5.6.7.8:88
kpasswd = kpasswd+tcp://1.2.3.4:464 kpasswd+tcp://5.6.7.8:464
```



重要

域配置参数必须列出由空格分隔的多个服务器，而不是像 /etc/krb5.conf 和 kdc.conf 那样，其中某些选项可以被多次指定。

3. 重启身份管理(IdM)服务：

```
# ipactl restart
```

其它资源

- 请参阅红帽知识库中的 为 AD Kerberos 通信将 IPA 服务器配置为 KDC 代理。

### 54.6. 配置 KKDCP 服务器 II

以下服务器配置依赖于 DNS 服务记录来查找要与之通信的活动目录(AD)服务器。

先决条件

- 您有 root 访问权限。

流程

1. 在 /etc/ipa/kdcproxy/kdcproxy.conf 文件中的 [global] 部分，将 use_dns 参数设为 true。

```
[global]
configs = mit
use_dns = true
```

configs 参数允许您加载其他配置模块。在这种情况下，配置是从 MIT libkrb5 库中读取的。

2. *可选* ：在您不想使用 DNS 服务记录的情况，请在 /etc/krb5.conf 文件的 [realms] 部分中添加明确的 AD 服务器。如果带有代理的域是 AD.*EXAMPLE.COM*，请添加：

```
[realms]
AD.EXAMPLE.COM = {
    kdc = ad-server.ad.example.com
    kpasswd_server = ad-server.ad.example.com
}
```

3. 重启身份管理(IdM)服务：

```
# ipactl restart
```

-

**其它资源**

- 请参阅红帽知识库中的为 AD Kerberos 通信将 IPA 服务器配置为 KDC 代理。

# 第 55 章 为 IDM 客户端上的 IDM 用户授予 SUDO 访问权限

了解有关在身份管理中授予用户 sudo 访问权限的更多信息。

## 55.1. IDM 客户端上的 SUDO 访问权限

系统管理员可以授予 sudo 访问权限，以允许非 root 用户执行通常为 root 用户保留的管理命令。因此，当用户需要执行通常为 root 用户保留的管理命令时，他们会在此命令前面使用 sudo。输入密码后，将像 root 用户一样执行 命令。要将 sudo 命令作为另一个用户或组（如数据库服务帐户）执行，您可以为 sudo 规则配置 *RunAs 别名*。

如果 Red Hat Enterprise Linux (RHEL) 8 主机注册为 Identity Management (IdM) 客户端，您可以指定 sudo 规则来定义哪些 IdM 用户可以在主机上执行哪些命令：

- 本地的 /etc/sudoers 文件中
- 集中在 IdM 中

您可以使用命令行界面(CLI)和 IdM Web UI 为 IdM 客户端创建 中央 sudo 规则。

在 RHEL 8.4 及更高版本中，您还可以使用通用安全服务应用程序编程接口 (GSSAPI) 为 sudo 配置免密码身份验证，这是基于 UNIX 的操作系统访问和验证 Kerberos 服务的本地方式。您可以使用 pam_sss_gss.so 可插拔验证模块 (PAM) 通过 SSSD 服务调用 GSSAPI 身份验证，允许用户通过有效的 Kerberos 票据向 sudo 命令进行身份验证。

其它资源

- 请参阅管理 sudo 访问。

## 55.2. 使用 CLI 向 IDM 客户端上的 IDM 用户授予 SUDO 访问权限

在 Identity Management (IdM) 中，您可以将特定命令的 sudo 访问权限授予特定 IdM 主机上的 IdM 用户帐户。首先，添加 sudo 命令，然后为一个或多个命令创建 sudo 规则。

例如，完成这个过程以创建 idm_user_reboot sudo 规则，为 idm_user 帐户授予在 idmclient 机器上

运行 /usr/sbin/reboot 命令的权限。

**先决条件**

- 以 IdM 管理员身份登录。
- 您已在 IdM 中创建 了 idm_user 的用户帐户，并通过为用户创建密码来解锁帐户。有关使用 CLI 添加新 IdM 用户的详情，请参阅使用命令行添加用户。
- idmclient 主机上没有本地的 idm_user。idm_user 用户未列在本地 /etc/passwd 文件中。

**流程**

1. 获取 Kerberos 票据作为 IdM admin。

```
[root@idmclient ~]# kinit admin
```

2. 在 sudo 命令的 IdM 数据库中添加 /usr/sbin/reboot 命令：

```
[root@idmclient ~]# ipa sudocmd-add /usr/sbin/reboot
----------------------------------
Added Sudo Command "/usr/sbin/reboot"
----------------------------------
  Sudo Command: /usr/sbin/reboot
```

3. 创建名为 idm_user_reboot 的 sudo 规则：

```
[root@idmclient ~]# ipa sudorule-add idm_user_reboot
--------------------------------
Added Sudo Rule "idm_user_reboot"
--------------------------------
  Rule name: idm_user_reboot
  Enabled: TRUE
```

4. 在 idm_user_reboot 规则中添加 /usr/sbin/reboot 命令：

```
[root@idmclient ~]# ipa sudorule-add-allow-command idm_user_reboot --sudocmds
'/usr/sbin/reboot'
  Rule name: idm_user_reboot
```

```
  Enabled: TRUE
  Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```

5. 将 idm_user_reboot 规则应用到 IdM idmclient 主机：

```
[root@idmclient ~]# ipa sudorule-add-host idm_user_reboot --hosts
idmclient.idm.example.com
Rule name: idm_user_reboot
Enabled: TRUE
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```

6. 在 idm_user_reboot 规则中添加 idm_user 帐户：

```
[root@idmclient ~]# ipa sudorule-add-user idm_user_reboot --users idm_user
Rule name: idm_user_reboot
Enabled: TRUE
Users: idm_user
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```

7. （可选）定义 idm_user_reboot 规则的有效性：

   a. 要定义 sudo 规则开始有效的时间，请使用带有 --setattr sudonotbefore=*DATE* 选项的 ipa sudorule-mod *sudo_rule_name* 命令。*DATE* 值必须遵循 yyyymmddHHMMSSZ 格式，以秒为单位。例如，要将 idm_user_reboot 规则的有效期的起始时间设为 2025 年 12 月 31 日 12:34:00，请输入：

```
[root@idmclient ~]# ipa sudorule-mod idm_user_reboot --setattr
sudonotbefore=20251231123400Z
```

   b. 要定义 sudo 规则停止有效期的时间，请使用 --setattr sudonotafter=DATE 选项。例如：要将 idm_user_reboot 规则有效期的截至时间设为 2026 年 12 月 31 日 12:34:00，请输入：

```
[root@idmclient ~]# ipa sudorule-mod idm_user_reboot --setattr
sudonotafter=20261231123400Z
```

**注意**

将更改从服务器传播到客户端可能需要几分钟时间。

**验证步骤**

1. 以 idm_user 帐户身份登录 idmclient 主机。

2. 显示允许 idm_user 帐户执行的 sudo 规则。

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idm_user on idmclient:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY KRB5CCNAME",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User idm_user may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

3. 使用 sudo 重新启动计算机。在提示时输入 idm_user 的密码：

```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

### 55.3. 使用 CLI 在 IDM 客户端上授予 SUDO 访问权限

身份管理(IdM)系统管理员可以使用 IdM 用户组设置访问权限、基于主机的访问控制、sudo 规则，以及 IdM 用户的其他控制。IdM 用户组授予并限制 IdM 域资源的访问权限。

您可以将活动目录(AD)*users* 和 AD *groups* 添加到 IdM 用户组中。要做到这一点：

1. 将 AD 用户或组添加到 *非 POSIX* 外部 IdM 组。

2. 将非 POSIX 外部 IdM 组添加到 IdM *POSIX* 组中。

然后，您可以通过管理 POSIX 组的权限来管理 AD 用户的权限。例如，您可以为特定 IdM 主机上的 IdM POSIX 用户组为特定命令授予 sudo 访问权限。

**注意**

也可以将 AD 用户组作为成员添加到 IdM 外部组中。通过使用户和组管理在一个 AD 域中，从而更轻松地为 Windows 用户定义策略。

**重要**

不要将 AD 用户的 ID 覆盖用于 IdM 中的 SUDO 规则。AD 用户的 ID 覆盖只代表 AD 用户的 POSIX 属性，而不是 AD 用户本身。

您可以作为组成员添加 ID 覆盖。但是，您只能使用此功能管理 IdM API 中的 IdM 资源。可以将 ID 覆盖添加为组群成员没有扩展到 POSIX 环境，因此您无法将其用于 sudo 或基于主机的访问控制(HBAC)规则中的成员资格。

按照以下流程创建 ad_users_reboot sudo 规则，为 administrator@ad-domain.com AD 用户授予在 idmclient IdM 主机上运行 /usr/sbin/reboot 命令的权限，其通常为 root 用户保留。administrator@ad-domain.com 是 ad_users_external 非 POSIX 组的成员，即 ad_users POSIX 组的成员。

**先决条件**

- 已获得 IdM 管理员 Kerberos ticket-granting ticket (TGT)。
- IdM 域和 ad-domain.com AD 域间存在跨林信任。
- idmclient 主机上没有本地 administrator 帐户：administrator 用户没有列在本地 /etc/passwd 文件中。

**流程**

1. **使用 administrator@ad-domain 成员 *创建包含 ad_users _external* 组的 ad_users 组：**

a. ***可选*** **：在 AD 域中创建或选择对应的组来管理 IdM 域中的 AD 用户。您可以使用多个 AD 组并将其添加到 IdM 端的不同组中。**

b. **创建 *ad_users_external* 组，并通过添加 --external 选项来指示来自 IdM 域以外的成员：**

```
[root@ipaserver ~]# ipa group-add --desc='AD users external map'
ad_users_external --external
-----------------------------
Added group "ad_users_external"
-----------------------------
  Group name: ad_users_external
  Description: AD users external map
```

**注意**

**确保此处指定的外部组是 AD 安全组，具有 global 或 universal 组范围，如 活动目录安全组 文档中所定义的。例如，无法使用 域用户 或域 管理员 AD 安全组，因为其组范围是 域本地。**

c. **创建 ad_users 组：**

```
[root@ipaserver ~]# ipa group-add --desc='AD users' ad_users
--------------------
Added group "ad_users"
--------------------
  Group name: ad_users
  Description: AD users
  GID: 129600004
```

d. **将 administrator@ad-domain.com AD 用户作为外部成员添加到 ad_users_external 中：**

```
[root@ipaserver ~]# ipa group-add-member ad_users_external --external
"administrator@ad-domain.com"
 [member user]:
 [member group]:
  Group name: ad_users_external
  Description: AD users external map
```

```
  External member: S-1-5-21-3655990580-1375374850-1633065477-513
-----------------------
Number of members added 1
-----------------------
```

AD 用户必须由完全限定名称来标识，如 DOMAIN\user_name 或 user_name@DOMAIN。然后，AD 身份映射到用户的 AD SID。添加 AD 组也是如此。

e. 将 ad_users_external 添加到 ad_users 作为成员：

```
[root@ipaserver ~]# ipa group-add-member ad_users --groups ad_users_external
  Group name: ad_users
  Description: AD users
  GID: 129600004
  Member groups: ad_users_external
-------------------------
Number of members added 1
-------------------------
```

2. 为 ad_users 的成员授予在 idmclient 主机上运行 /usr/sbin/reboot 的权限：

a. 在 sudo 命令的 IdM 数据库中添加 /usr/sbin/reboot 命令：

```
[root@idmclient ~]# ipa sudocmd-add /usr/sbin/reboot
-----------------------------------
Added Sudo Command "/usr/sbin/reboot"
-----------------------------------
  Sudo Command: /usr/sbin/reboot
```

b. 创建名为 ad_users_reboot 的 sudo 规则：

```
[root@idmclient ~]# ipa sudorule-add ad_users_reboot
-------------------------------
Added Sudo Rule "ad_users_reboot"
-------------------------------
  Rule name: ad_users_reboot
  Enabled: True
```

c. 在 ad_users_reboot 规则中添加 /usr/sbin/reboot 命令：

```
[root@idmclient ~]# ipa sudorule-add-allow-command ad_users_reboot --
sudocmds '/usr/sbin/reboot'
  Rule name: ad_users_reboot
```

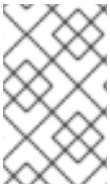
```
  Enabled: True
  Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```

d. 将 ad_users_reboot 规则应用到 IdM idmclient 主机：

```
[root@idmclient ~]# ipa sudorule-add-host ad_users_reboot --hosts
idmclient.idm.example.com
Rule name: ad_users_reboot
Enabled: True
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```

e. 将 ad_users 组添加到 ad_users_reboot 规则中：

```
[root@idmclient ~]# ipa sudorule-add-user ad_users_reboot --groups ad_users
Rule name: ad_users_reboot
Enabled: TRUE
User Groups: ad_users
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```



**注意**

将更改从服务器传播到客户端可能需要几分钟时间。

**验证步骤**

1. 以 administrator@ad-domain.com 身份登录 idmclient 主机，它是 ad_users 组的间接成员：

```
$ ssh administrator@ad-domain.com@ipaclient
Password:
```

2. 另外，显示 administrator@ad-domain.com 允许执行的 sudo 命令：

```
[administrator@ad-domain.com@idmclient ~]$ sudo -l
Matching Defaults entries for administrator@ad-domain.com on idmclient:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY KRB5CCNAME",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User administrator@ad-domain.com may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

3. 使用 sudo 重新启动计算机。提示时输入 administrator@ad-domain.com 的密码：

```
[administrator@ad-domain.com@idmclient ~]$ sudo /usr/sbin/reboot
[sudo] password for administrator@ad-domain.com:
```

**其它资源**

- Active Directory 用户和身份管理组
- 将可信 Active Directory 域中的用户和组包含到 SUDO 规则

## 55.4. 使用 IDM WEB UI 向 IDM 客户端上的 IDM 用户授予 SUDO 访问权限

在 Identity Management (IdM) 中，您可以将特定命令的 sudo 访问权限授予特定 IdM 主机上的 IdM 用户帐户。首先，添加 sudo 命令，然后为一个或多个命令创建 sudo 规则。

完成此步骤以创建 idm_user_reboot sudo 规则，为 idm_user 帐户授予在 idmclient 计算机上运行 /usr/sbin/reboot 命令的权限。

**先决条件**

- 以 IdM 管理员身份登录。
- 您已在 IdM 中创建 了 idm_user 的用户帐户，并通过为用户创建密码来解锁帐户。有关使用

命令行界面添加新 IdM 用户的详情，请参阅使用命令行添加用户。

- idmclient 主机上没有本地的 idm_user。idm_user 用户未列在本地 /etc/passwd 文件中。

**流程**

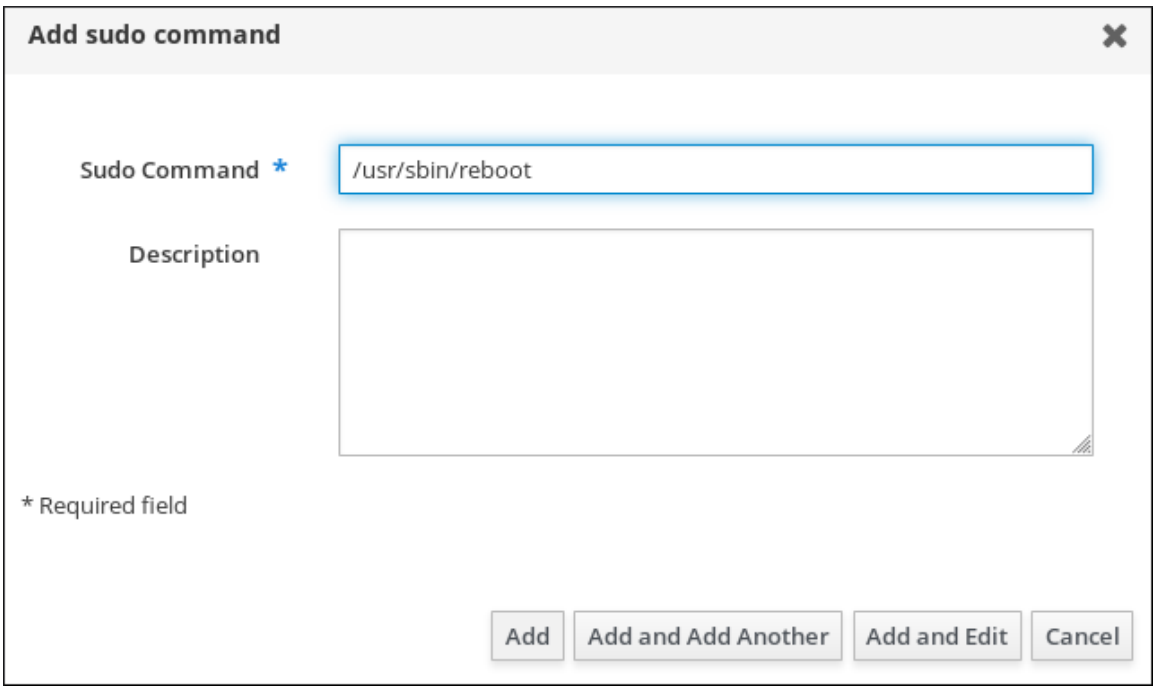
1. 在 sudo 命令的 IdM 数据库中添加 /usr/sbin/reboot 命令：

   a. 导航到 Policy → Sudo → Sudo Commands。

   b. 单击右上角的 Add，以打开 Add sudo 命令对话框。

   c. 输入您希望用户能够使用 sudo 执行的命令：/usr/sbin/reboot。

   图 55.1. 添加 IdM sudo 命令




   d. 点击 Add。

2. 使用新的 sudo 命令条目创建一个 sudo 规则来允许 idm_user 重启 idmclient 机器：

a. 导航到 Policy → Sudo → Sudo rules。

b. 单击右上角的 Add，以打开 Add sudo 规则对话框。

c. 输入 sudo 规则的名称： idm_user_reboot。

d. 点 Add and Edit。

e. 指定用户：

i. 在 Who 部分中，选中指定的用户和组单选按钮。

ii. 在 User category the rule applies to 子小节中，点 Add 打开 Add users into sudo rule "idm_user_reboot" 对话框。

iii. 在 Available 栏的 Add users into sudo rule "idm_user_reboot" 对话框中，选择 idm_user，并把它移到 Prospective 栏。

iv. 点击 Add。

f. 指定主机：

i. 在 Access this host 部分中，选中指定的 Hosts and Groups 单选按钮。

ii. 在 Host category this rule applies to 子小节中，点 Add 打开 Add hosts into sudo rule "idm_user_reboot" 对话框。

iii. 在 Available 列中的 Add hosts to sudo rule "idm_user_reboot" 对话框中，选中 idmclient.idm.example.com 复选框，并将它移到 Prospective 列。

iv. 点击 **Add**。

g. 指定命令：

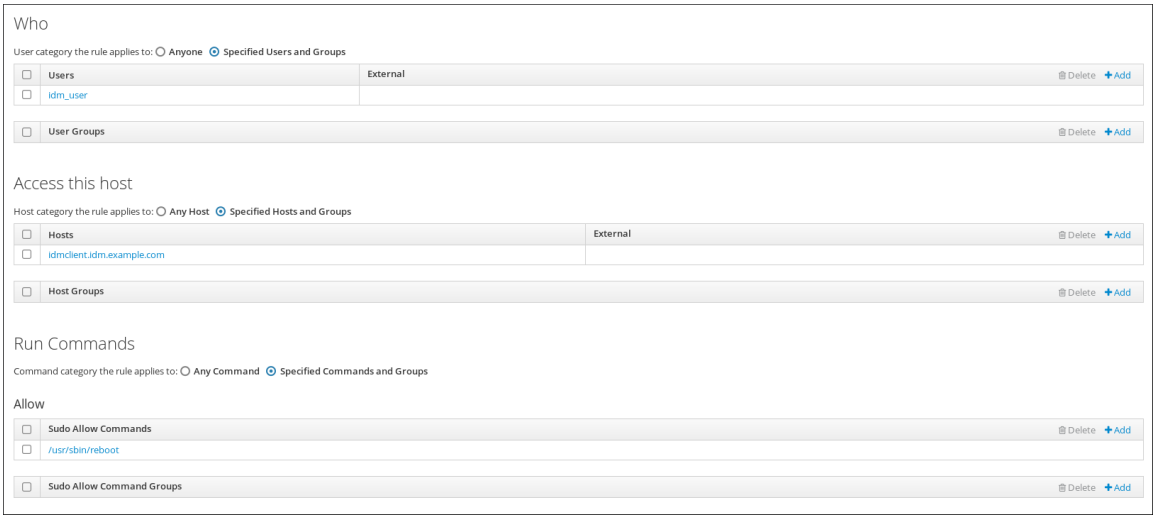
i. 在 **Run Commands** 一节的 **Command category the rule applies to** 子小节中，选择 **Specified Commands and Groups** 单选按钮。

ii. 在 **Sudo Allow Commands** 子节中，单击 **Add** 以打开 **Add allow sudo commands into sudo rule "idm_user_reboot"**对话框。

iii. 在 **Available** 列中的 **Add allow sudo commands into sudo rule "idm_user_reboot"** 对话框中，选中 **/usr/sbin/reboot** 复选框，并将它移到 **Prospective** 列。

iv. 点 **Add** 返回到 **idm_sudo_reboot** 页。

**图 55.2. 添加 IdM sudo 规则**




h. 单击左上角的 **Save**。

新规则默认为启用。

**注意**

将更改从服务器传播到客户端可能需要几分钟时间。

**验证步骤**

1. 以 idm_user 用户身份登录 idmclient。

2. 使用 sudo 重新启动计算机。在提示时输入 idm_user 的密码：

```
$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

如果正确配置了 sudo 规则，机器将重启。

## 55.5. 在 CLI 上创建 SUDO 规则，以作为 IDM 客户端上的服务帐户运行命令

在 IdM 中，您可以使用 *RunAs alias* 配置 sudo 规则，以便以另一个用户或组身份运行 sudo 命令。例如，您可能有一个托管数据库应用的 IdM 客户端，您需要以与该应用对应的本地服务帐户运行命令。

使用此示例在命令行上创建名为 run_third-party-app_report 的 sudo 规则，以允许 idm_user 帐户以 idmclient 主机上 thirdpartyapp 服务帐户的身份运行/opt/third-party-app/bin/report 命令。

**先决条件**

- 以 IdM 管理员身份登录。
- 您已在 IdM 中创建了 idm_user 的用户帐户，并通过为用户创建密码来解锁帐户。有关使用 CLI 添加新 IdM 用户的详情，请参阅使用命令行添加用户。
- idmclient 主机上没有本地的 idm_user。idm_user 用户未列在本地 /etc/passwd 文件中。
- 您有一个名为 third-party-app 的自定义应用程序安装在 idmclient 主机上。

- third-party-app 应用程序的report 命令安装在 /opt/third-party-app/bin/report 目录中。
- 您已创建了一个名为 thirdpartyapp 的本地服务帐户，来为 third-party-app 应用程序执行命令。

**流程**

1. 获取 Kerberos 票据作为 IdM admin。

```
[root@idmclient ~]# kinit admin
```

2. 将 /opt/third-party-app/bin/report 命令添加到 sudo 命令的 IdM 数据库：

```
[root@idmclient ~]# ipa sudocmd-add /opt/third-party-app/bin/report
----------------------------------------------------
Added Sudo Command "/opt/third-party-app/bin/report"
----------------------------------------------------
  Sudo Command: /opt/third-party-app/bin/report
```

3. 创建一个名为 run_third-party-app_report 的 sudo 规则：

```
[root@idmclient ~]# ipa sudorule-add run_third-party-app_report
-------------------------------------------
Added Sudo Rule "run_third-party-app_report"
-------------------------------------------
  Rule name: run_third-party-app_report
  Enabled: TRUE
```

4. 使用 --users=*<user>* 选项来为 sudorule-add-runasuser 命令指定 RunAs 用户：

```
[root@idmclient ~]# ipa sudorule-add-runasuser run_third-party-app_report --
users=thirdpartyapp
  Rule name: run_third-party-app_report
  Enabled: TRUE
  RunAs External User: thirdpartyapp
-----------------------
Number of members added 1
-----------------------
```

用户（或使用 --groups=* 选项指定 的组）可以是 IdM 的外部用户，如本地服务帐户或活动目录用户。不要为组名称添加 % 前缀。

5. 将 /opt/third-party-app/bin/report 命令添加到 run_third-party-app_report 规则中：

```
[root@idmclient ~]# ipa sudorule-add-allow-command run_third-party-app_report --
sudocmds '/opt/third-party-app/bin/report'
Rule name: run_third-party-app_report
Enabled: TRUE
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
-------------------------
Number of members added 1
-------------------------
```

6. 将 run_third- party-app_report 规则应用到 IdM idmclient 主机：

```
[root@idmclient ~]# ipa sudorule-add-host run_third-party-app_report --hosts
idmclient.idm.example.com
Rule name: run_third-party-app_report
Enabled: TRUE
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
-------------------------
Number of members added 1
-------------------------
```

7. 将 idm_user 帐户添加到 run_third- party-app_report 规则中：

```
[root@idmclient ~]# ipa sudorule-add-user run_third-party-app_report --users
idm_user
Rule name: run_third-party-app_report
Enabled: TRUE
Users: idm_user
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
-------------------------
Number of members added 1
```

**注意**

将更改从服务器传播到客户端可能需要几分钟时间。

验证步骤

1. 以 idm_user 帐户身份登录 idmclient 主机。

2. 测试新的 sudo 规则：

   a. 显示允许 idm_user 帐户执行的 sudo 规则。

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idm_user@idm.example.com on idmclient:
    !visiblepw, always_set_home, match_group_by_gid,
always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS
LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY KRB5CCNAME",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User idm_user@idm.example.com may run the following commands on idmclient:
    (thirdpartyapp) /opt/third-party-app/bin/report
```

   b. 作为 thirdpartyapp 服务帐户，运行 report 命令。

```
[idm_user@idmclient ~]$ sudo -u thirdpartyapp /opt/third-party-app/bin/report
[sudo] password for idm_user@idm.example.com:
Executing report...
Report successful.
```

## 55.6. 在 IDM WEB UI 中创建一个 SUDO 规则，该规则在 IDM 客户端上以服务帐户的身份运行命令

在 IdM 中，您可以使用 *RunAs alias* 配置 sudo 规则，以便以另一个用户或组身份运行 sudo 命令。例如，您可能有一个托管数据库应用的 IdM 客户端，您需要以与该应用对应的本地服务帐户运行命令。

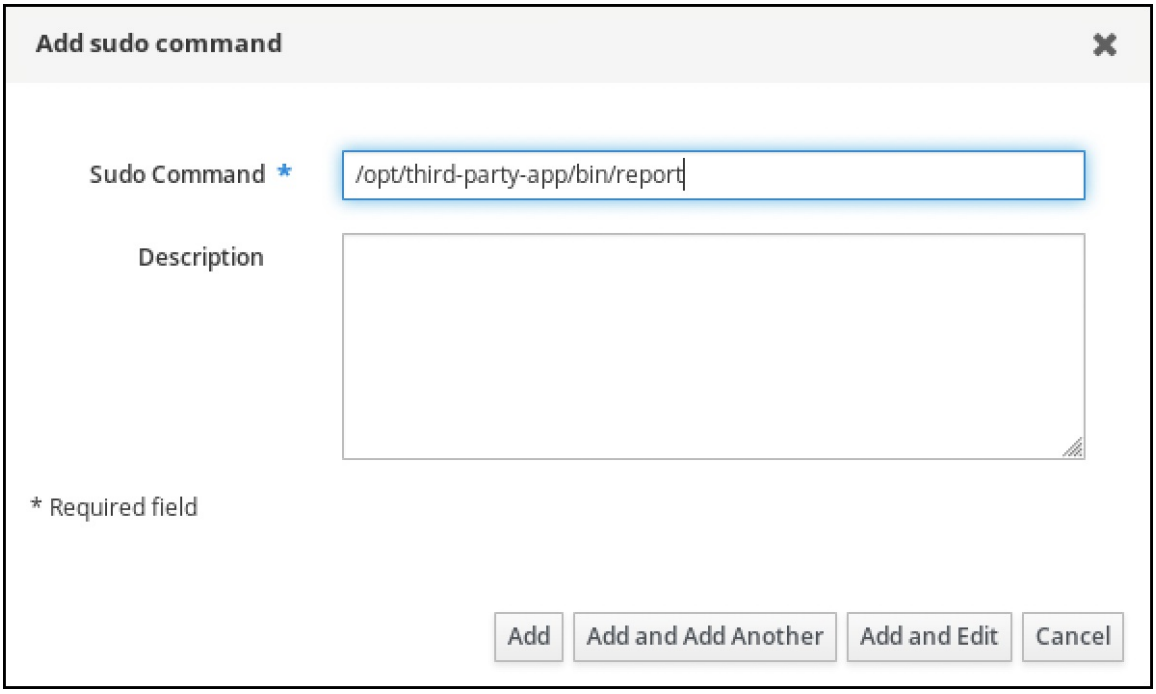
使用此示例在 IdM Web UI 中创建一个名为 run_third-party-app_report 的 sudo 规则，以允许 idm_user 帐户以 idmclient 主机上 thirdpartyapp 服务账号的身份运行 /opt/third-party-app/bin/report 命令。

**先决条件**

- 以 IdM 管理员身份登录。
- 您已在 IdM 中创建 了 idm_user 的用户帐户，并通过为用户创建密码来解锁帐户。有关使用 CLI 添加新 IdM 用户的详情，请参阅使用命令行添加用户。
- idmclient 主机上没有本地的 idm_user。idm_user 用户未列在本地 /etc/passwd 文件中。
- 您有一个名为 third-party-app 的自定义应用程序安装在 idmclient 主机上。
- third-party-app 应用程序的report 命令安装在 /opt/third-party-app/bin/report 目录中。
- 您已创建了一个名为 thirdpartyapp 的本地服务帐户，来为 third-party-app 应用程序执行命令。

**流程**

1. 将 /opt/third-party-app/bin/report 命令添加到 sudo 命令的 IdM 数据库：
   a. 导航到 Policy → Sudo → Sudo Commands。
   b. 单击右上角的 Add，以打开 Add sudo 命令对话框。
   c. 输入命令：/opt/third-party-app/bin/report。



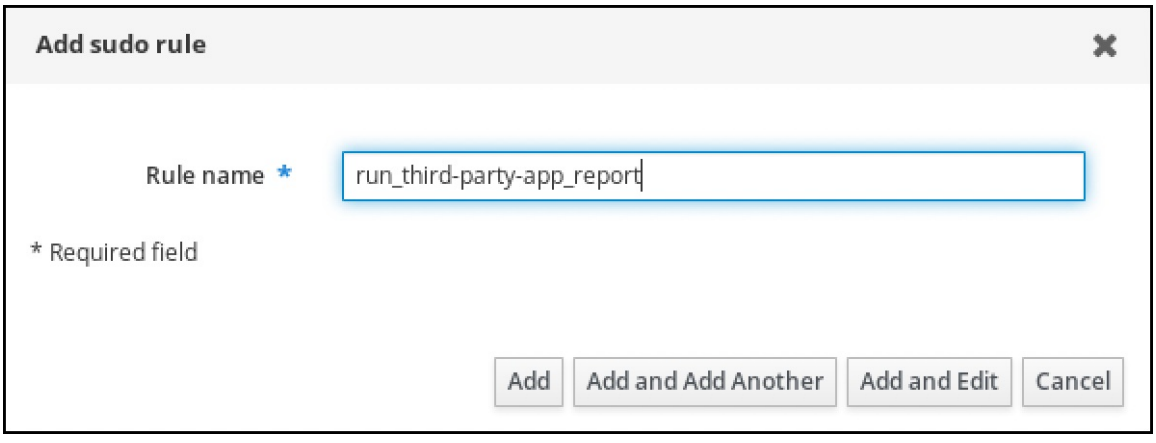

d. **点击 Add。**

2. **使用新的 sudo 命令条目来创建新的 sudo 规则：**

a. **导航到 Policy → Sudo → Sudo rules。**

b. **单击右上角的 Add，以打开 Add sudo 规则对话框。**

c. **输入 sudo 规则的名称：run_third-party-app_report。**



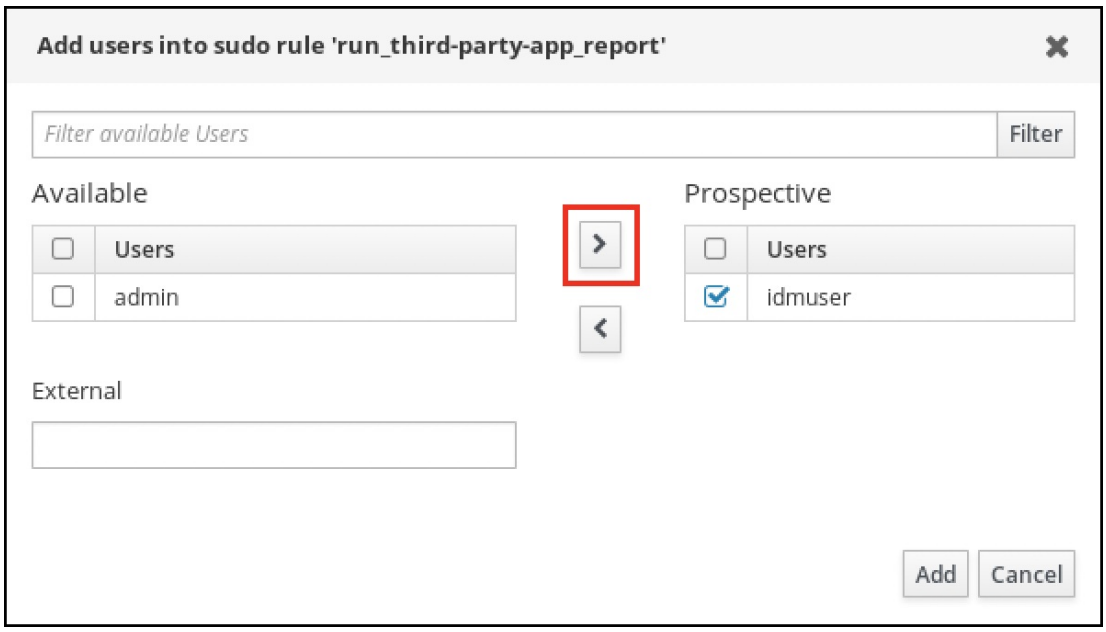

d. **点 Add and Edit。**

e. **指定用户：**

i. **在 Who 部分中，选中指定的用户和组单选按钮。**

ii. **在 规则应用到的用户类别 子部分，单击 Add 来打开 将用户添加到 sudo 规则 "run_third-party-app_report" 对话框。**

iii. **在 Available 栏的 Add users into sudo rule "run_third-party-app_report" 对话框中，选择 idm_user，并把它移到 Prospective 栏。**



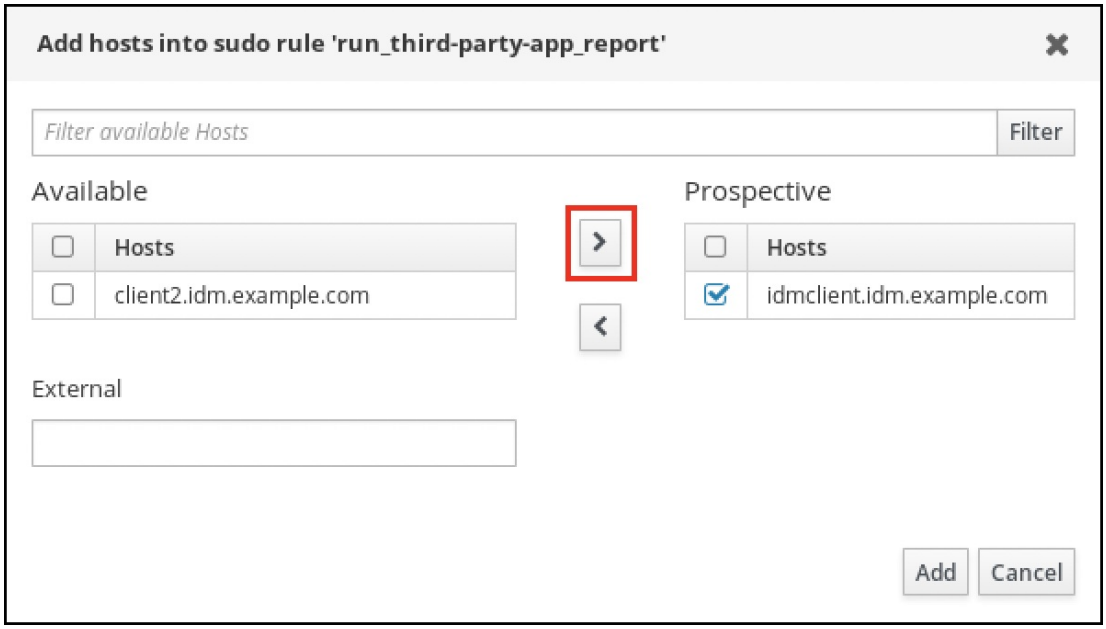

iv. **点击 Add。**

f. **指定主机：**

i. **在 Access this host 部分中，选中指定的 Hosts and Groups 单选按钮。**

ii. **在 此规则应用到主机类别 子部分中，单击 Add 来打开 将主机添加到 sudo 规则 "run_third- third-app_report" 对话框。**

iii. **在 Available 栏的 Add hosts to sudo rule "run_third- party-app_report" 对话框中，选中 idmclient.idm.example.com 复选框，并将它移到 Prospective 列。**



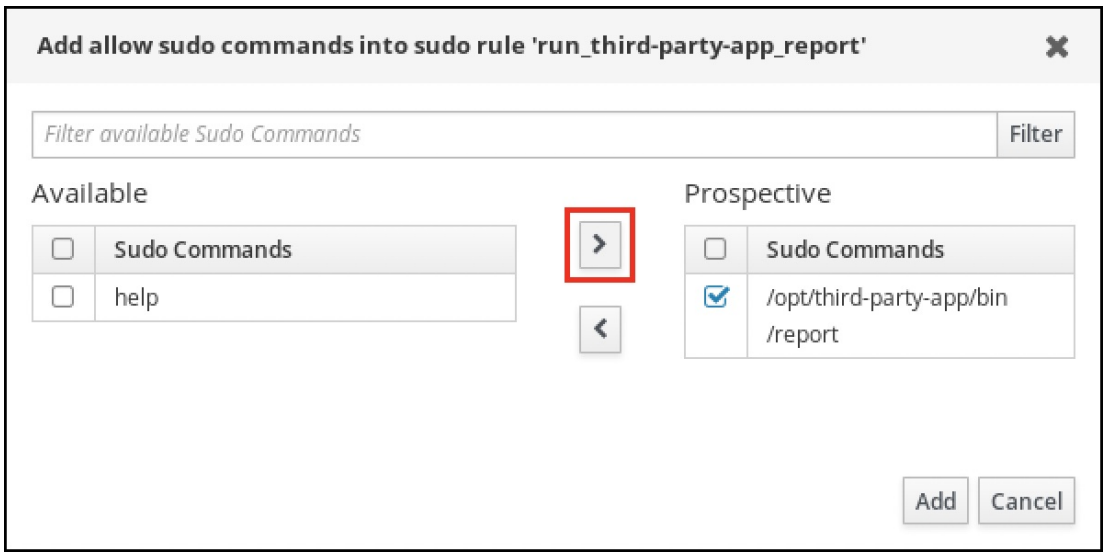

iv. **点击 Add。**

g. **指定命令：**

i. **在 Run Commands 一节的 Command category the rule applies to 子小节中，选择 Specified Commands and Groups 单选按钮。**

ii. **在 Sudo 允许的命令 子部分中，单击 Add 来打开 将允许的 sudo 命令添加到 sudo 规则 "run_third-app_report" 对话框。**

iii. **在 Available 栏的 Add allow sudo commands into sudo rule "run_third-party-app_report" 对话框中，选中 /opt/third-party-app/bin/report 并将其移到 Prospective 栏。**



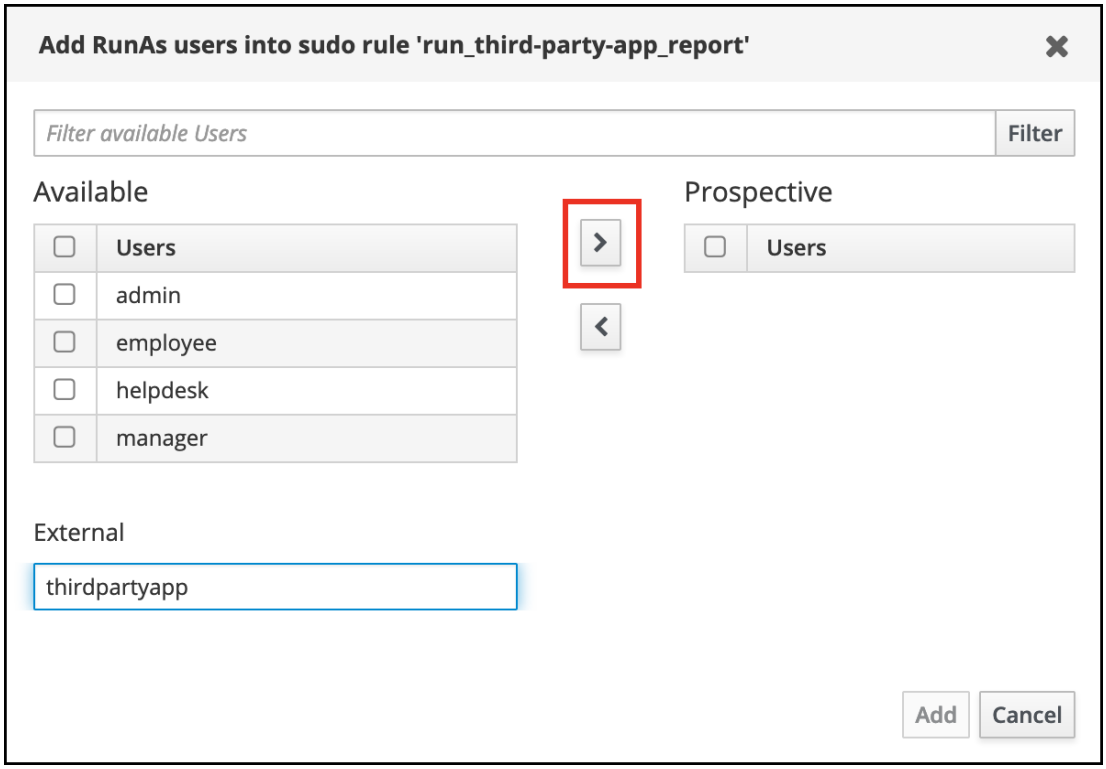
iv. 单击 **Add** 以返回到 **run_third-party-app_report** 页面。

h. **指定 RunAs 用户：**

i. **在 As Whom 部分中，选中 指定的用户和组 单选按钮。**

ii. **在 RunAs Users 子部分中，单击 Add 以打开 将 RunAs 用户添加到 sudo 规则 "run_third-app_report" 对话框。**

iii. **在 将 RunAs 用户添加到 sudo 规则 "run_third-app_report" 对话框中，在 External 框中输入 thirdpartyapp 服务帐户，并将其移到 Prospective 列中。**

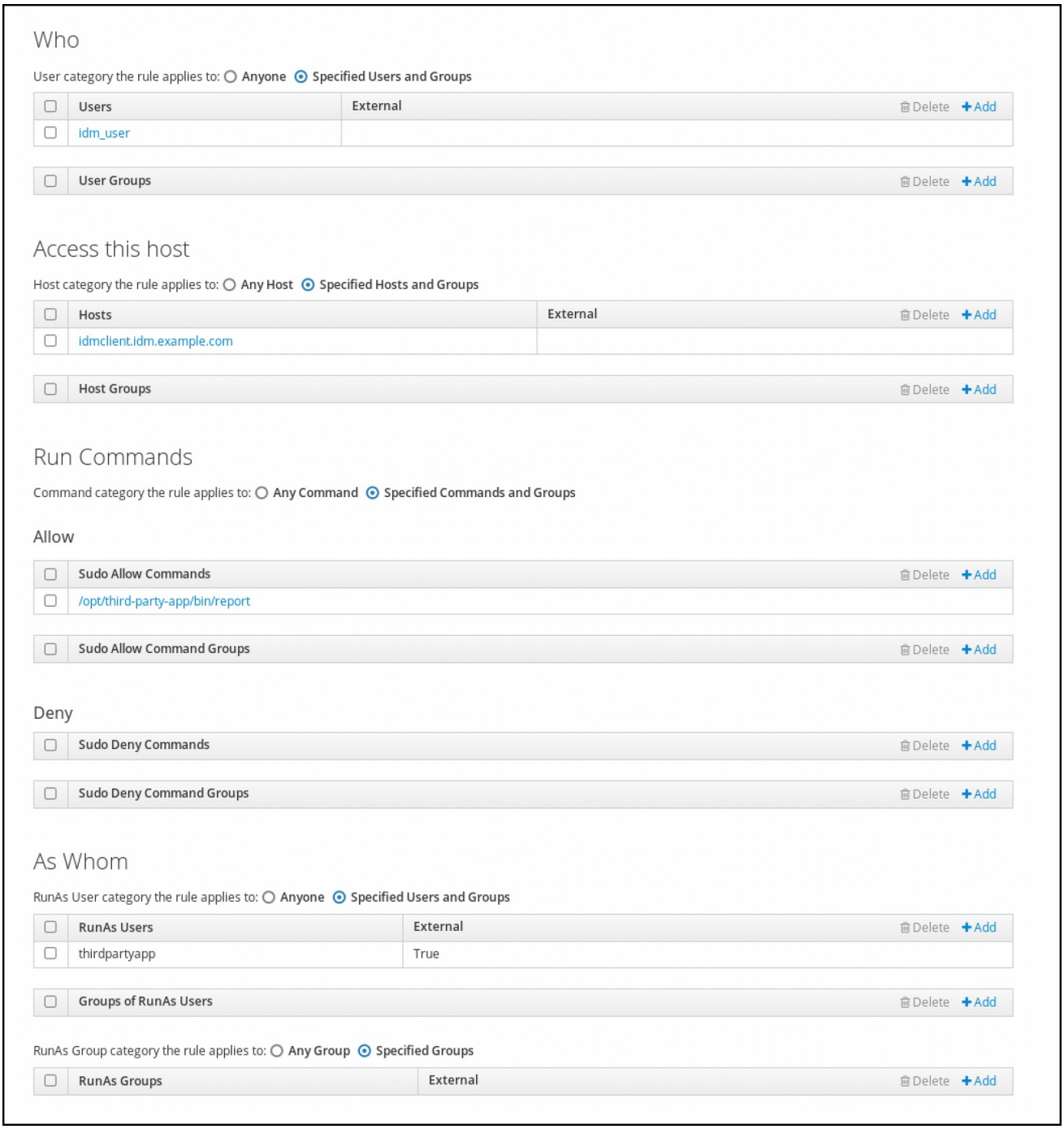


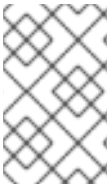

iv. 单击 **Add** 以返回到 **run_third-party-app_report** 页面。

i. **单击左上角的 Save。**

**新规则默认为启用。**

图 55.3. sudo 规则的详情



**注意**

**将更改从服务器传播到客户端可能需要几分钟时间。**

**验证步骤**

1. **以 idm_user 帐户身份登录 idmclient 主机。**

2. 测试新的 sudo 规则：

a. 显示允许 idm_user 帐户执行的 sudo 规则。

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idm_user@idm.example.com on idmclient:
    !visiblepw, always_set_home, match_group_by_gid,
always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS
LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY KRB5CCNAME",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User idm_user@idm.example.com may run the following commands on idmclient:
    (thirdpartyapp) /opt/third-party-app/bin/report
```

b. 作为 thirdpartyapp 服务帐户，运行 report 命令。

```
[idm_user@idmclient ~]$ sudo -u thirdpartyapp /opt/third-party-app/bin/report
[sudo] password for idm_user@idm.example.com:
Executing report...
Report successful.
```

## 55.7. 在 IDM 客户端上为 SUDO 启用 GSSAPI 身份验证

以下流程描述了通过 pam_sss_gss.so PAM 模块在 IdM 客户端上为 sudo 和 sudo -i 命令启用通用安全服务应用程序接口(GSSAPI)身份验证。使用这个配置，IdM 用户可以使用它们的 Kerberos 票据对 sudo 命令进行身份验证。

**先决条件**

- 您已为应用于 IdM 主机的 IdM 用户创建了 sudo 规则。在本例中，您已创建了 idm_user_reboot sudo 规则，为 idm_user 帐户授予在 idmclient 主机上运行 /usr/sbin/reboot 命令的权限。
- idmclient 主机正在运行 RHEL 8.4 或更高版本。

- 您需要 root 权限来修改 /etc/sssd/sssd.conf 文件和 /etc/pam.d/ 目录中的 PAM 文件。

**流程**

1. 打开 /etc/sssd/sssd.conf 配置文件：

2. 在 [domain/*<domain_name>* ] 部分中添加以下条目。

```
[domain/<domain_name>]
pam_gssapi_services = sudo, sudo-i
```

3. 保存并关闭 /etc/sssd/sssd.conf 文件。

4. 重启 SSSD 服务以载入配置更改。

```
[root@idmclient ~]# systemctl restart sssd
```

5. 如果您正在运行 RHEL 8.8 或更高版本：

   a. [可选] 确定是否您已选择了 sssd authselect 配置文件：

   ```
   # authselect current
   Profile ID: sssd
   ```

   输出显示选择了 sssd authselect 配置文件。

   b. 如果选择了 sssd authselect 配置文件，请启用 GSSAPI 身份验证：

   ```
   # authselect enable-feature with-gssapi
   ```

   c. 如果没有选择 sssd authselect 配置文件，请选择它并启用 GSSAPI 身份验证：

   ```
   # authselect select sssd with-gssapi
   ```

6. 如果您正在运行 RHEL 8.7 或更早版本：

   a. 打开 /etc/pam.d/sudo PAM 配置文件。

   b. 添加下列条目，作为 /etc/pam.d/sudo 文件中的 auth 部分的第一行。

```
#%PAM-1.0
auth sufficient pam_sss_gss.so
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
```

   c. 保存并关闭 /etc/pam.d/sudo 文件。

验证步骤

1. 以 idm_user 帐户身份登录到主机。

```
[root@idm-client ~]# ssh -l idm_user@idm.example.com localhost
idm_user@idm.example.com's password:
```

2. 验证您有一个票据授予票据作为 idm_user 帐户。

```
[idmuser@idmclient ~]$ klist
Ticket cache: KCM:1366201107
Default principal: idm_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
01/08/2021 09:11:48  01/08/2021 19:11:48
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
 renew until 01/15/2021 09:11:44
```

3. *（可选）* 如果您没有 idm_user 帐户的 Kerberos 凭证，请删除您当前的 Kerberos 凭证，并请求正确的凭证。

```
[idm_user@idmclient ~]$ kdestroy -A

[idm_user@idmclient ~]$ kinit idm_user@IDM.EXAMPLE.COM
Password for idm_user@idm.example.com:
```

4. 使用 sudo 重启机器，而不用指定密码。

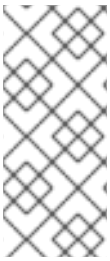
```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
```

其它资源

- IdM 术语 列表中的 GSSAPI 条目
- 使用 IdM Web UI，授予 sudo 访问 IdM 客户端上 IdM 用户的权限
- 使用 CLI 向 IdM 客户端上的 IdM 用户授予 sudo 访问权限
- pam_sss_gss(8) 手册页
- sssd.conf(5) 手册页

## 55.8. 在 IDM 客户端上为 SUDO 启用 GSSAPI 身份验证，并强制实施 KERBEROS 身份验证指标

以下流程描述了通过 pam_sss_gss.so PAM 模块在 IdM 客户端上为 sudo 和 sudo -i 命令启用通用安全服务应用程序接口(GSSAPI)身份验证。此外，只有已使用智能卡登录的用户才能使用 他们的 Kerberos 票据对这些命令进行身份验证。



注意

您可以将此流程作为模板，使用 SSSD 为其他 PAM 感知的服务配置 GSSAPI 身份验证，并进一步限制只对那些在其 Kerberos 票据上附加了特定身份验证指标的用户进行访问。

先决条件

- 您已为应用于 IdM 主机的 IdM 用户创建了 sudo 规则。在本例中，您已创建了 idm_user_reboot sudo 规则，为 idm_user 帐户授予在 idmclient 主机上运行 /usr/sbin/reboot 命令的权限。

- 您已为 idmclient 主机配置了智能卡身份验证。
- idmclient 主机正在运行 RHEL 8.4 或更高版本。
- 您需要 root 权限来修改 /etc/sssd/sssd.conf 文件和 /etc/pam.d/ 目录中的 PAM 文件。

**流程**

1. 打开 /etc/sssd/sssd.conf 配置文件：
2. 将以下条目添加到 [domain/*<domain_name>*] 部分中。

```
[domain/<domain_name>]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:pkinit, sudo-i:pkinit
```

3. 保存并关闭 /etc/sssd/sssd.conf 文件。
4. 重启 SSSD 服务以载入配置更改。

```
[root@idmclient ~]# systemctl restart sssd
```

5. 打开 /etc/pam.d/sudo PAM 配置文件。
6. 添加下列条目，作为 /etc/pam.d/sudo 文件中的 auth 部分的第一行。

```
#%PAM-1.0
auth sufficient pam_sss_gss.so
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
```

7. 保存并关闭 /etc/pam.d/sudo 文件。

8. 打开 /etc/pam.d/sudo-i PAM 配置文件。

9. 添加下列条目，作为 /etc/pam.d/sudo-i 文件中的 auth 部分的第一行。

```
#%PAM-1.0
auth sufficient pam_sss_gss.so
auth       include      sudo
account    include      sudo
password   include      sudo
session    optional     pam_keyinit.so force revoke
session    include      sudo
```

10. 保存并关闭 /etc/pam.d/sudo-i 文件。

验证步骤

1. 以 idm_user 帐户登录到主机，并使用智能卡进行身份验证。

```
[root@idmclient ~]# ssh -l idm_user@idm.example.com localhost
PIN for smart_card
```

2. 验证作为智能卡用户，您有一个票据授予票据。

```
[idm_user@idmclient ~]$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: idm_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
02/15/2021 16:29:48  02/16/2021 02:29:48
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
 renew until 02/22/2021 16:29:44
```

3. 显示允许 idm_user 帐户执行的 sudo 规则。

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idmuser on idmclient:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
```

```
LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY KRB5CCNAME",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User idm_user may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

4. 使用 sudo 重启机器，而不用指定密码。

```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
```

其它资源

- SSSD 选项控制对 PAM 服务的 GSSAPI 身份验证
- IdM 术语 列表中的 GSSAPI 条目
- 为智能卡验证配置身份管理
- Kerberos 认证指示符
- 使用 IdM Web UI，授予 sudo 访问 IdM 客户端上 IdM 用户的权限
- 使用 CLI 向 IdM 客户端上的 IdM 用户授予 sudo 访问权限。
- pam_sss_gss(8) 手册页
- sssd.conf(5) 手册页

## 55.9. SSSD 选项控制对 PAM 服务的 GSSAPI 身份验证

您可以对 /etc/sssd/sssd.conf 配置文件使用以下选项来调整 SSSD 服务中的 GSSAPI 配置。

pam_gssapi_services

默认情况下，禁用带有 SSSD 的 GSSAPI 身份验证。您可以使用此选项来指定一个以逗号分隔的 PAM 服务列表，允许这些服务使用 pam_sss_gss.gss.so PAM 模块尝试 GSSAPI 身份验证。要显式禁用 GSSAPI 身份验证，将这个选项设为 -。

pam_gssapi_indicators_map

这个选项只适用于身份管理(IdM)域。使用这个选项列出授予 PAM 访问服务所需的 Kerberos 身份验证指标。配对的格式必须是 *<PAM_service>:* _<required_authentication_indicator>_。

有效的验证指标为：

- OTP 用于双因素身份验证
- radius 用于 RADIUS 身份验证
- pkinit 用于 PKINIT、智能卡或证书身份验证
- hardened 用于强化的密码

pam_gssapi_check_upn

默认启用这个选项，并将其设为 true。如果启用了这个选项，SSSD 服务要求用户名与 Kerberos 凭证匹配。如果为 false，pam_ss_gss.so PAM 模块将对能够获取所需服务票据的每个用户进行身份验证。

示例

以下选项为 sudo 和 sudo-i 服务启用 Kerberos 身份验证，要求 sudo 用户使用一次性密码进行身份验证，用户名必须与 Kerberos 主体匹配。由于这些设置位于 [pam] 部分中，因此适用于所有域：

```
[pam]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:otp
pam_gssapi_check_upn = true
```

您还可以在单独的 [domain] 部分中设置这些选项，来覆盖 [pam] 部分中的任何全局值。以下选项对每个域应用不同的 GSSAPI 设置：

对于 idm.example.com 域

- 为 sudo 和 sudo -i 服务启用 GSSAPI 身份验证。
- sudo 命令需要证书或智能卡身份验证器。
- sudo -i 命令需要一次性密码身份验证器.
- 强制匹配用户名和 Kerberos 主体.

对于 ad.example.com 域

- 仅为 sudo 服务启用 GSSAPI 身份验证。
- 不强制匹配用户名和主体。

```
[domain/idm.example.com]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:pkinit, sudo-i:otp
pam_gssapi_check_upn = true
...

[domain/ad.example.com]
pam_gssapi_services = sudo
pam_gssapi_check_upn = false
...
```

其它资源

- Kerberos 认证指示符

### 55.10. SUDO 的 GSSAPI 身份验证故障排除

如果您无法使用 IdM 的 Kerberos 票据对 sudo 服务进行身份验证，请使用以下场景对您的配置进行故障排除。

先决条件

- 您已为 sudo 服务启用了 GSSAPI 身份验证。请参阅 在 IdM 客户端上为 sudo 启用 GSSAPI

身份验证。

- 您需要 root 权限来修改 /etc/sssd/sssd.conf 文件和 /etc/pam.d/ 目录中的 PAM 文件。

流程

- 如果您看到以下错误，Kerberos 服务可能无法为基于主机名的服务票据解析正确的域：

```
Server not found in Kerberos database
```

在这种情况下，将主机名直接添加到 /etc/krb5.conf Kerberos 配置文件中的 [domain_realm] 部分：

```
[idm-user@idm-client ~]$ cat /etc/krb5.conf
...

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
 server.example.com = EXAMPLE.COM
```

- 如果看到以下错误，则您没有任何 Kerberos 凭证：

```
No Kerberos credentials available
```

在这种情况下，使用 kinit 工具检索 Kerberos 凭证，或者通过 SSSD 进行身份验证：

```
[idm-user@idm-client ~]$ kinit idm-user@IDM.EXAMPLE.COM
Password for idm-user@idm.example.com:
```

- 如果您在 /var/log/sssd/sssd_pam.log 日志文件中看到以下错误之一，则 Kerberos 凭证与当前登录的用户的用户名不匹配：

```
User with UPN [<UPN>] was not found.

UPN [<UPN>] does not match target user [<username>].
```

在这种情况下，验证您使用 SSSD 进行身份验证，或考虑禁用 /etc/sssd/sssd.conf 文件中的 pam_gssapi_check_upn 选项：

```
[idm-user@idm-client ~]$ cat /etc/sssd/sssd.conf
...

pam_gssapi_check_upn = false
```

- 若要进行额外的故障排除，您可以对 pam_sss_gss.so PAM 模块启用调试输出。

  - 在 PAM 文件（如 /etc/pam.d/sudo 和 /etc/pam.d/sudo-i ）中所有 pam_sss_gss.so 条目的末尾添加 debug 选项：

```
[root@idm-client ~]# cat /etc/pam.d/sudo
#%PAM-1.0
auth       sufficient   pam_sss_gss.so   debug
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
```

```
[root@idm-client ~]# cat /etc/pam.d/sudo-i
#%PAM-1.0
auth       sufficient   pam_sss_gss.so   debug
auth       include      sudo
account    include      sudo
password   include      sudo
session    optional     pam_keyinit.so force revoke
session    include      sudo
```

  - 尝试使用 pam_sss_gss.so 模块进行身份验证，并查看控制台输出。在本例中，用户没有任何 Kerberos 凭据。

```
[idm-user@idm-client ~]$ sudo ls -l /etc/sssd/sssd.conf
pam_sss_gss: Initializing GSSAPI authentication with SSSD
pam_sss_gss: Switching euid from 0 to 1366201107
pam_sss_gss: Trying to establish security context
pam_sss_gss: SSSD User name: idm-user@idm.example.com
pam_sss_gss: User domain: idm.example.com
pam_sss_gss: User principal:
pam_sss_gss: Target name: host@idm.example.com
pam_sss_gss: Using ccache: KCM:
pam_sss_gss: Acquiring credentials, principal name will be derived
pam_sss_gss: Unable to read credentials from [KCM:] [maj:0xd0000,
min:0x96c73ac3]
pam_sss_gss: GSSAPI: Unspecified GSS failure.  Minor code may provide more
information
pam_sss_gss: GSSAPI: No credentials cache found
pam_sss_gss: Switching euid from 1366200907 to 0
pam_sss_gss: System error [5]: Input/output error
```

## 55.11. 使用 ANSIBLE PLAYBOOK 确保 IDM 客户端上的 IDM 用户具有 SUDO 访问权限

在身份管理(IdM)中，您可以确保对特定命令的 sudo 访问权限被授予给特定 IdM 主机上的 IdM 用户帐户。

完成此流程以确保名为 idm_user_reboot 的 sudo 规则存在。该规则授予 idm_user 在 idmclient 机器上运行 /usr/sbin/reboot 命令的权限。

### 先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您已 确保 IdM 中存在 idm_user 用户帐户，并通过为用户创建密码解锁了帐户。有关使用命令行界面添加新 IdM 用户的详情，请参考链接： 使用命令行 添加用户。
- idmclient 中没有本地 idm_user 帐户。idm_user 用户未列在 idmclient 上的 /etc/passwd 文件中。

### 流程

1. 创建一个清单文件，如 inventory.file，并在其中定义 ipaservers ：

```
[ipaservers]
server.idm.example.com
```

2. 添加一个或多个 sudo 命令：

   a. 创建一个 ensure-reboot-sudocmd-is-present.yml Ansible playbook，来确保 sudo 命令的 IdM 数据库中存在 /usr/sbin/reboot 命令。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/sudocmd/ensure-sudocmd-is-present.yml 文件中的示例：

```
---
- name: Playbook to manage sudo command
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure sudo command is present
  - ipasudocmd:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: /usr/sbin/reboot
      state: present
```

   b. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
reboot-sudocmd-is-present.yml
```

3. 创建一个引用命令的 sudo 规则：

   a. 创建一个 ensure-sudorule-for-idmuser-on-idmclient-is-present.yml Ansible playbook，来使用 sudo 命令条目确保存在 sudo 规则。sudo 规则允许 idm_user 重新启动 idmclient 机器。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/sudorule/ensure-sudorule-is-present.yml 文件中的示例：

```
---
- name: Tests
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure a sudorule is present granting idm_user the permission to run
/usr/sbin/reboot on idmclient
```

```
- ipasudorule:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: idm_user_reboot
    description: A test sudo rule.
    allow_sudocmd: /usr/sbin/reboot
    host: idmclient.idm.example.com
    user: idm_user
    state: present
```

b. 运行 playbook：

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-sudorule-for-idmuser-on-idmclient-is-
present.yml
```

验证步骤

通过验证 idm_user 能够使用 sudo 重启 idmclient，来测试您在 IdM 服务器上确认其存在性的 sudo 规则是否在 idmclient 上可以工作。请注意，可能需要过几分钟后，服务器上所做的更改才会对客户端生效。

1. 以 idm_user 用户身份登录到 idmclient。

2. 使用 sudo 重新启动计算机。在提示时输入 idm_user 的密码：

```
$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

如果正确配置了 sudo，则机器将重启。

其它资源

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-sudocmd.md ,README-sudocmdgroup.md 和 README-sudorule.md 文件。

# 第 56 章 配置基于主机的访问控制规则

您可以使用基于主机的访问控制(HBAC)规则来管理身份管理(IdM)域中的访问控制。HBAC 规则定义哪些用户或用户组可以使用服务组中的哪些服务或服务访问指定的主机或主机组。例如，您可以使用 HBAC 规则来实现以下目标：

- 将您域中对指定系统的访问权限限制为特定用户组的成员。
- 仅允许使用特定的服务来访问域中的系统。

默认情况下，使用名为 allow_all 的默认 HBAC 规则配置 IdM，该规则允许用户通过整个 IdM 域中的每个相关服务对每个主机进行通用访问。

您可以通过将默认的 allow_all 规则替换为您自己的一组 HBAC 规则来微调对不同主机的访问。对于集中式和简化的访问控制管理，您可以将 HBAC 规则应用到用户组、主机组或服务组，而不是单个用户、主机或服务。

## 56.1. 使用 WEBUI 在 IDM 域中配置 HBAC 规则

要为基于主机的访问控制配置域，请完成以下步骤：

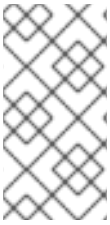
1. 在 IdM WebUI 中创建 HBAC 规则。
2. 测试新的 HBAC 规则。
3. 禁用默认的 allow_all HBAC 规则。

**注意**

在创建自定义 HBAC 规则前不要禁用 allow_all 规则，因为这样做了，任何用户将无法访问任何主机。

### 56.1.1. 在 IdM WebUI 中创建 HBAC 规则

要使用 IdM Web UI 为基于主机的访问控制配置域，请按照以下步骤操作。出于本示例的目的，流程演示了如何授予单个用户 *sysadmin* 使用任何服务访问域中的所有系统。



**注意**

IdM 将用户的主组存储为 gidNumber 属性的数字值，而不是到 IdM 组对象的链接。因此，HBAC 规则只能引用用户的补充组，而不是其主组。

**先决条件**

- 用户 *sysadmin* 在 IdM 中存在。

**流程**

1. 选择 **Policy>Host-Based Access Control>HBAC Rules**。
2. 点 **Add** 开始添加新规则。
3. 输入规则的名称，然后点 **Add and Edit** 打开 HBAC 规则配置页面。
4. 在 **Who** 区域中，选择 **Specified Users and Groups**。然后点 **Add** 添加用户或组。
5. 从 **Available** 用户列表中选择 *sysadmin* 用户，点击 **>** 进入到 **Prospective** 用户列表，然后点击 **Add**。
6. 在 **Accessing** 区域中，选择 **Any Host** 来将 HBAC 规则应用到所有主机。
7. 在 **Via Service** 区域中，选择 **Any Service** 来将 HBAC 规则应用到所有服务。

**注意**

默认情况下，只为 HBAC 规则配置最常见的服务和服务组。

- 要显示当前可用的服务的列表，请选择 **Policy>Host-Based Access Control>HBAC Services**。
- 要显示当前可用的服务组的列表，请选择 **Policy>Host-Based Access Control>HBAC Service Groups**。

要添加更多服务和服务组，请参阅 为自定义 HBAC 服务添加 HBAC 服务条目 和 添加 HBAC 服务组。

8. 要保存您在 HBAC rule 配置页面上所做的任何更改，请点击页面顶部的 **Save**。

### 56.1.2. 在 IdM WebUI 中测试 HBAC 规则

IdM 允许您使用模拟场景测试各种情况下的 HBAC 配置。执行这些模拟测试，您可以在生产环境中部署 HBAC 规则前发现错误配置问题或安全风险。

**重要**

在生产环境中开始使用它们之前，请始终测试自定义 HBAC 规则。

请注意，IdM 不测试 HBAC 规则对可信活动目录(AD)用户的影响。因为 IdM LDAP 目录不存储 AD 数据，所以当模拟 HBAC 场景时，IdM 无法解析 AD 用户的组成员资格。

**流程**

1. 选择 **Policy>Host-Based Access Control>HBAC Test**。
2. 在 **Who** 窗口中，指定您要在其下执行测试的用户，然后点 **Next**。

3. 在 Accessing 窗口中，指定用户将尝试访问的主机，然后单击 Next。

4. 在 Via Service 窗口上，指定用户将尝试使用的服务，然后单击 Next。

5. 在 Rules 窗口中，选择您要测试的 HBAC 规则，然后点 Next。如果您没有选择任何规则，则会测试所有规则。

选择 Include Enabled 来针对状态为 Enabled 的所有规则运行测试。选择 Include Disabled 来针对状态为 Disabled 的所有规则运行测试。要查看并更改 HBAC 规则的状态，请选择 Policy>Host-Based Access Control>HBAC Rules。

重要

如果对多个规则运行测试，如果至少一个所选规则允许访问，则成功通过。

6. 在 Run Test 窗口上，单击 Run Test。

7. 查看测试结果：

- 如果您看到 ACCESS DENIED，则用户在测试中没有授予访问权限。
- 如果您看到 ACCESS GRANTED，该用户可以成功访问主机。

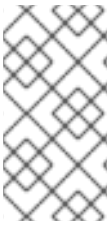
默认情况下，IdM 在显示测试结果时会列出所有经过测试的 HBAC 规则。

- 选择 Matched 以显示允许成功访问的规则。
- 选择 Unmatched 来显示阻止访问的规则。

### 56.1.3. 在 IdM WebUI 中禁用 HBAC 规则

您可以禁用 HBAC 规则，但它只停用该规则，不会删除它。如果禁用了一个 HBAC 规则，您可以稍后

重新启用它。



**注意**

当您首次配置自定义 HBAC 规则时，禁用 HBAC 规则很有用。要确保新配置没有被默认的 allow_all HBAC 规则覆盖，您必须禁用 allow_all。

**流程**

1. 选择 **Policy>Host-Based Access Control>HBAC Rules**。
2. 选择您要禁用的 HBAC 规则。
3. 单击 **Disable**。
4. 点 OK 以确认您要禁用所选的 HBAC 规则。

## 56.2. 在 IDM 域中使用 CLI 配置 HBAC 规则

要为基于主机的访问控制配置域，请完成以下步骤：

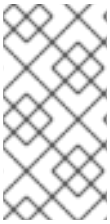
1. 在 IdM CLI 中创建 HBAC 规则。
2. 测试新的 HBAC 规则。
3. 禁用默认的 allow_all HBAC 规则。

**注意**

在创建自定义 HBAC 规则前，不要禁用 allow_all 规则。如果您在创建自定义规则前禁用了它，则所有用户对所有主机的访问都将被拒绝。

### 56.2.1. 在 IdM CLI 中创建 HBAC 规则

要使用 IdM CLI 为基于主机的访问控制配置域，请按照以下步骤操作。出于本示例的目的，流程展示了如何授予单个用户 *sysadmin* 使用任何服务访问域中所有系统的权限。



**注意**

IdM 将用户的主组存储为 gidNumber 属性的数字值，而不是到 IdM 组对象的链接。因此，HBAC 规则只能引用用户的补充组，而不是其主组。

**先决条件**

- 用户 *sysadmin* 在 IdM 中存在。

**流程**

1. 使用 ipa hbacrule-add 命令添加规则。

```
$ ipa hbacrule-add
Rule name: rule_name
---------------------------
Added HBAC rule "rule_name"
---------------------------
  Rule name: rule_name
  Enabled: TRUE
```

2. 要将 HBAC 规则只应用到 *sysadmin* 用户，请使用 ipa hbacrule-add-user 命令。

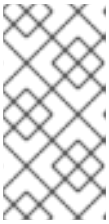
```
$ ipa hbacrule-add-user --users=sysadmin
Rule name: rule_name
  Rule name: rule_name
  Enabled: True
  Users: sysadmin
-------------------------
Number of members added 1
-------------------------
```

**注意**

要将 HBAC 规则应用到所有用户，请使用 ipa hbacrule-mod 命令，并指定所有用户类别 --usercat=all。请注意，如果 HBAC 规则与单个用户或组关联，ipa hbacrule-mod --usercat=all 会失败。在这种情况下，使用 ipa hbacrule-remove-user 命令删除用户和组。

3. 指定目标主机。要将 HBAC 规则应用到所有主机，请使用 ipa hbacrule-mod 命令，并指定所有主机类别：

```
$ ipa hbacrule-mod rule_name --hostcat=all
----------------------------
Modified HBAC rule "rule_name"
----------------------------
  Rule name: rule_name
  Host category: all
  Enabled: TRUE
  Users: sysadmin
```

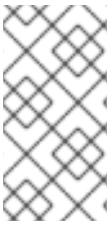


**注意**

如果 HBAC 规则与单个主机或组关联，ipa hbacrule-mod --hostcat=all 失败。在这种情况下，使用 ipa hbacrule-remove-host 命令删除主机和组。

4. 指定目标 HBAC 服务。要将 HBAC 规则应用到所有服务，请使用 ipa hbacrule-mod 命令，并指定所有服务类别：

```
$ ipa hbacrule-mod rule_name --servicecat=all
----------------------------
Modified HBAC rule "rule_name"
----------------------------
  Rule name: rule_name
  Host category: all
  Service category: all
  Enabled: True
  Users: sysadmin
```



**注意**

如果 HBAC 规则与单个服务或组关联，ipa hbacrule-mod --servicecat=all 失败。在这种情况下，使用 ipa hbacrule-remove-service 命令删除服务和组。

验证

- 验证 HBAC 规则是否已正确添加。

  a. 使用 ipa hbacrule-find 命令验证 HBAC 规则是否在 IdM 中存在。

b. 使用 ipa hbacrule-show 命令验证 HBAC 规则的属性。

**其它资源**

- 如需了解更多详细信息，请参阅 ipa hbacrule-add --help。
- 请参阅 为自定义 HBAC 服务添加 HBAC 服务条目。
- 请参阅 添加 HBAC 服务组。

### 56.2.2. 在 IdM CLI 中测试 HBAC 规则

IdM 允许您使用模拟场景测试各种情况下的 HBAC 配置。执行这些模拟测试，您可以在生产环境中部署 HBAC 规则前发现错误配置问题或安全风险。

在生产环境中开始使用它们之前，请始终测试自定义 HBAC 规则。

请注意，IdM 不测试 HBAC 规则对可信活动目录(AD)用户的影响。因为 IdM LDAP 目录不存储 AD 数据，所以当模拟 HBAC 场景时，IdM 无法解析 AD 用户的组成员资格。

**流程**

1. 使用 ipa hbactest 命令测试您的 HBAC 规则。您有测试单个 HBAC 规则或多个 HBAC 规则的选项。

   - 要测试单个 HBAC 规则：

     ```
     $ ipa hbactest --user=sysadmin --host=server.idm.example.com --service=sudo --
     rules=rule_name
     ----------------------
     Access granted: True
     ----------------------
       Matched rules: rule_name
     ```

   - 要测试多个 HBAC 规则：

a. 添加一个仅允许 *sysadmin* 在所有主机上使用 ssh的第二个规则：

```
$ ipa hbacrule-add --hostcat=all rule2_name
$ ipa hbacrule-add-user --users sysadmin rule2_name
$ ipa hbacrule-add-service --hbacsvcs=sshd rule2_name
  Rule name: rule2_name
  Host category: all
  Enabled: True
  Users: admin
  HBAC Services: sshd
-------------------------
Number of members added 1
-------------------------
```

b. 运行以下命令来测试多个 HBAC 规则：

```
$ ipa hbactest --user=sysadmin --host=server.idm.example.com --
service=sudo --rules=rule_name --rules=rule2_name
--------------------
Access granted: True
--------------------
  Matched rules: rule_name
  Not matched rules: rule2_name
```

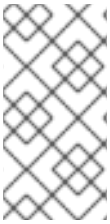
在输出中，Matched rules 列出了允许成功访问的规则，而 Not matched 规则列出了阻止访问的规则。请注意，如果您没有指定 --rules 选项，则会应用所有规则。--rules 可用于单独测试每个规则。

**其它资源**

- 如需更多信息，请参阅 ipa hbactest --help。

### 56.2.3. 在 IdM CLI 中禁用 HBAC 规则

您可以禁用 HBAC 规则，但它只停用该规则，不会删除它。如果禁用了一个 HBAC 规则，您可以稍后重新启用它。



**注意**

当您首次配置自定义 HBAC 规则时，禁用 HBAC 规则很有用。要确保新配置没有被默认的 allow_all HBAC 规则覆盖，您必须禁用 allow_all。

**流程**

- 使用 ipa hbacrule-disable 命令。例如，要禁用 allow_all 规则：

```
$ ipa hbacrule-disable allow_all
-----------------------------
Disabled HBAC rule "allow_all"
-----------------------------
```

**其它资源**

- 如需了解更多详细信息，请参阅 ipa hbacrule-disable --help。

### 56.3. 为自定义 HBAC 服务添加 HBAC 服务条目

默认为 HBAC 规则配置最常见的服务和服务组，但您也可以将任何其他可插拔身份验证模块(PAM)服务配置为 HBAC 服务。这允许您在 HBAC 规则中定义自定义 PAM 服务。这些 PAM 服务文件位于 RHEL 系统上的 etc/pam.d 目录中。

**注意**

将服务添加为 HBAC 服务与向域添加服务不同。向域中添加服务使其可用于域中的其他资源，但不允许在 HBAC 规则中使用该服务。

#### 56.3.1. 在 IdM Web UI 中为自定义 HBAC 服务添加 HBAC 服务条目

要添加一个自定义 HBAC 服务条目，请按照以下描述的步骤操作。

**流程**

1. 选择 Policy>Host-Based Access Control>HBAC Services。
2. 点 Add 添加一个 HBAC 服务条目。
3. 输入服务的名称，然后单击 Add。

#### 56.3.2. 在 IdM CLI 中为自定义 HBAC 服务添加 HBAC 服务条目

要添加一个自定义 HBAC 服务条目，请按照以下描述的步骤操作。

**流程**

- 使用 ipa hbacsvc-add 命令。例如，要为 tftp 服务添加一个条目：

```
$ ipa hbacsvc-add tftp
-----------------------
Added HBAC service "tftp"
-----------------------
 Service name: tftp
```

**其它资源**

- 如需了解更多详细信息，请参阅 ipa hbacsvc-add --help。

### 56.4. 添加 HBAC 服务组

HBAC 服务组可以简化 HBAC 规则管理。例如，您可以添加整个服务组，而不是将单个服务添加到 HBAC 规则中。

#### 56.4.1. 在 IdM WebUI 中添加 HBAC 服务组

要在 IdM WebUI 中添加一个 HBAC 服务组，请按照以下步骤操作。

**流程**

1. 选择 Policy>Host-Based Access Control>HBAC Service Groups。
2. 点 Add 添加 HBAC 服务组。
3. 输入服务组的名称，然后点 Edit。
4. 在服务组配置页面中，点 Add 将 HBAC 服务添加为组的成员。

#### 56.4.2. 在 IdM CLI 中添加 HBAC 服务组

要在 IdM CLI 中添加一个 HBAC 服务组，请按照以下步骤操作。

**流程**

1. 在终端中使用 ipa hbacsvcgroup-add 命令添加一个 HBAC 服务组。例如，要添加名为 *login* 的组：

```
$ ipa hbacsvcgroup-add
Service group name: login
------------------------------
Added HBAC service group "login"
------------------------------
  Service group name: login
```

2. 使用 ipa hbacsvcgroup-add-member 命令，将 HBAC 服务添加为组的成员。例如，要将 sshd 服务添加到 *login* 组中：

```
$ ipa hbacsvcgroup-add-member
Service group name: login
[member HBAC service]: sshd
  Service group name: login
  Member HBAC service: sshd
------------------------
Number of members added 1
------------------------
```

**其它资源**

- 如需了解更多详细信息，请参阅 ipa hbacsvcgroup-add --help。
- 如需了解更多详细信息，请参阅 ipa hbacsvcgroup-add-member --help。

# 第 57 章 确保使用 ANSIBLE PLAYBOOK 的基于主机的访问控制规则在 IDM 中存在

Ansible 是一个自动化工具，用于配置系统、部署软件和执行滚动更新。它包括对身份管理(IdM)的支持。

了解更多有关基于主机的访问策略的信息，以及如何使用 Ansible 定义它们。

## 57.1. IDM 中基于主机的访问控制规则

基于主机的访问控制(HBAC)规则定义哪些用户或用户组可以通过哪些服务或服务组中的哪些服务来访问哪些主机或主机组。作为系统管理员，您可以使用 HBAC 规则来实现以下目标：

- 将您域中对指定系统的访问权限限制为特定用户组的成员。
- 仅允许使用特定服务来访问域中的系统。

默认情况下，IdM 是使用一个名为 allow_all 的默认 HBAC 规则配置的，这意味着每个用户都可以通过整个 IdM 域中每个相关服务对每个主机进行通用访问。

您可以通过将默认的 allow_all 规则替换为您自己的一组 HBAC 规则来微调对不同主机的访问。对于集中式和简化的访问控制管理，您可以将 HBAC 规则应用到用户组、主机组或服务组，而不是单个用户、主机或服务。

## 57.2. 使用 ANSIBLE PLAYBOOK 确保在 IDM 中存在 HBAC 规则

按照以下流程，使用 Ansible playbook 确保基于主机的访问控制(HBAC)规则在身份管理(IdM)中存在。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。

- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- IdM 中存在您要用于 HBAC 规则的用户和用户组。详情请参阅 使用 Ansible playbook 管理用户帐户 ，以及 使用 Ansible playbook 确保 IdM 组和组成员存在。
- 您要应用 HBAC 规则的主机和主机组在 IdM 中存在。详情请参阅 使用 Ansible playbook 管理主机，以及 使用 Ansible playbook 管理主机组。

**流程**

1. 创建一个清单文件，如 inventory.file，并在该文件中定义 ipaserver ：

```
[ipaserver]
server.idm.example.com
```

2. 创建 Ansible playbook 文件，该文件定义您要确保其存在的 HBAC 策略。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/hbacrule/ensure-hbacrule-allhosts-present.yml 文件中的示例：

```
---
- name: Playbook to handle hbacrules
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure idm_user can access client.idm.example.com via the sshd service
  - ipahbacrule:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: login
      user: idm_user
      host: client.idm.example.com
```

```
    hbacsvc:
    - sshd
    state: present
```

3. 运行 playbook：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-new-
hbacrule-present.yml
```

**验证步骤**

1. 以管理员身份登录到 IdM Web UI。
2. 导航到 **Policy** → **Host-Based-Access-Control** → **HBAC Test**。
3. 在 **Who** 选项卡中，选择 **idm_user**。
4. 在 **Accessing** 选项卡中，选择 **client.idm.example.com**。
5. 在 **Via service** 选项卡中，选择 **sshd**。
6. 在 **Rules** 选项卡中，选择 **login**。
7. 在 **Run test** 选项卡中，单击 **Run test** 按钮。如果您看到 **ACCESS GRANTED**，则 HBAC 规则成功实现。

**其它资源**

- 请参阅 **/usr/share/doc/ansible-freeipa** 目录中的 **README-hbacsvc.md** , **README-hbacsvcgroup.md** 和 **README-hbacrule.md** 文件。
- 请参阅 **/usr/share/doc/ansible-freeipa/playbooks** 目录的子目录中的 playbook。

# 第 58 章 管理复制拓扑

本章论述了如何管理身份管理(IdM)域中服务器之间的复制。

其它资源

- 规划副本拓扑

## 58.1. 解释复制协议、拓扑后缀和拓扑片段

当您创建副本时，身份管理(IdM)会在初始服务器和副本之间创建一个复制协议。然后，复制的数据会存储在拓扑后缀中，当两个副本在它们的后缀之间有复制协议时，后缀会形成一个拓扑段。在以下部分中更为详细地解释了这些概念：

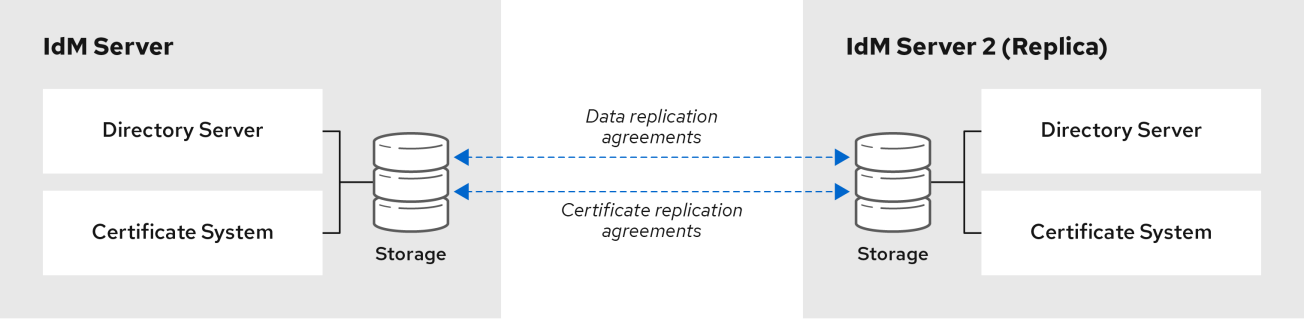
- 复制协议
- 拓扑后缀
- 拓扑段

### 58.1.1. IdM 副本之间的复制协议

当管理员基于现有服务器创建副本时，身份管理 (IdM) 会在初始服务器和副本之间创建*复制协议*。复制协议确保两个服务器之间不断复制数据和配置。

IdM 使用*多读/写副本复制*。在这种配置中，所有副本都加入到复制协议中接收并提供更新，因此被视为供应商和消费者。复制协议始终是强制的。

图 58.1. 服务器和副本协议




IdM 使用两种复制协议：

域复制协议

这些协议复制身份信息。

证书复制协议

这些协议复制证书信息。

两个复制频道都是独立的。两个服务器可以有一类或两种类型的复制协议。例如，当服务器 A 和服务器 B 仅配置了域复制协议时，它们之间仅复制身份信息，而不复制证书信息。

### 58.1.2. 拓扑后缀

*拓扑后缀*存储复制的数据。IdM 支持两种类型的拓扑后缀：domain 和 ca。每个后缀代表一个单独的服务器，即一个单独的复制拓扑。

配置复制协议时，它会在两个不同的服务器上加入同一类型的两个拓扑后缀。

域 后缀：dc=*示例*,dc=*com*

域 后缀包含与域相关的所有数据。

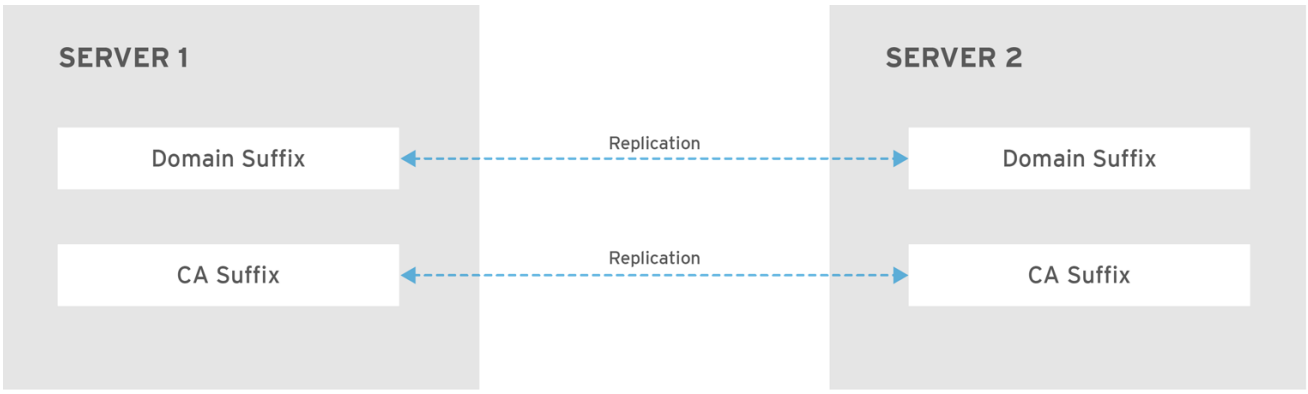
当两个副本在其 域 后缀之间具有复制协议时，它们共享目录数据，如用户、组和策略。

ca 后缀：o=ipaca

ca 后缀包含证书系统组件的数据。它仅存在于安装有证书颁发机构 (CA) 的服务器上。

当两个副本在其 ca 后缀之间具有复制协议时，它们会共享证书数据。

图 58.2. 拓扑后缀




在安装新副本时，ipa- replica-install 脚本会在两个服务器之间设置初始拓扑复制协议。

例 58.1. 查看拓扑后缀

ipa topologysuffix-find 命令显示拓扑后缀列表：

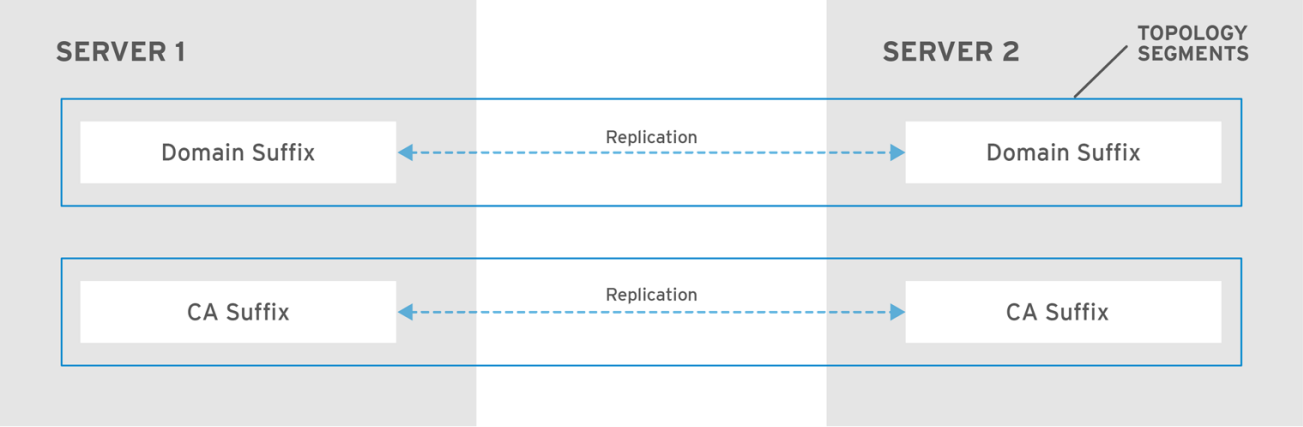
```
$ ipa topologysuffix-find
---------------------------
2 topology suffixes matched
---------------------------
  Suffix name: ca
  Managed LDAP suffix DN: o=ipaca

  Suffix name: domain
  Managed LDAP suffix DN: dc=example,dc=com
----------------------------
Number of entries returned 2
----------------------------
```

### 58.1.3. 拓扑片段

当两个副本在其后缀之间具有复制协议时，后缀组成一个 *拓扑片段*。每个拓扑段由一个 *左节点和一个右节点* 组成。节点代表复制协议中加入的服务器。

IdM 中的拓扑片段始终是双向的。每个部分代表两种复制协议：从服务器 A 到服务器 B，从服务器 B 复制到服务器 A。因此数据会同时复制到服务器 A。

图 58.3. 拓扑片段




例 58.2. 查看拓扑片段

ipa topologysegment-find 命令显示为域或 CA 后缀配置的当前拓扑片段。例如，对于域后缀：

```
$ ipa topologysegment-find
Suffix name: domain
-----------------
1 segment matched
-----------------
  Segment name: server1.example.com-to-server2.example.com
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both
----------------------------
Number of entries returned 1
----------------------------
```

在本例中，域相关数据仅在两个服务器之间复制：server1.example.com 和 server2.example.com 。

要只显示特定片段的详情，请使用 ipa topologysegment-show 命令：

```
$ ipa topologysegment-show
Suffix name: domain
Segment name: server1.example.com-to-server2.example.com
  Segment name: server1.example.com-to-server2.example.com
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both
```

## 58.2. 使用拓扑图管理复制拓扑

Web UI 中的拓扑图显示域中服务器之间的关系。您可以使用 Web UI 来操作和转换拓扑表示法。

### 访问拓扑图

访问拓扑图：

1. 选择 IPA Server → Topology → Topology Graph。
2. 如果您对拓扑进行任何没有立即反映在图形中的更改，点 Refresh。

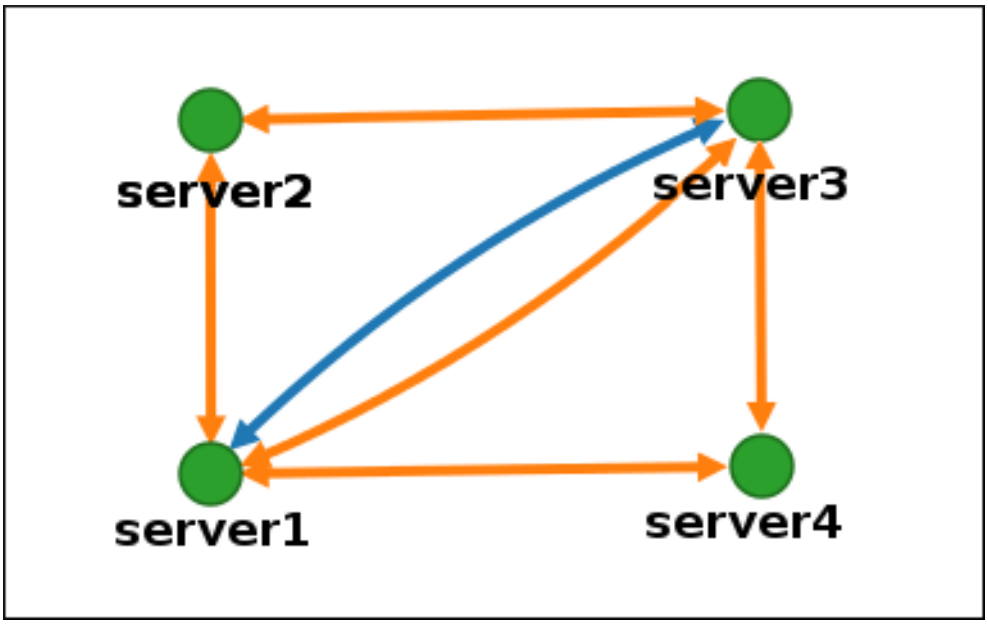
### 解读拓扑图

加入域复制协议的服务器通过圆形箭头连接。加入 CA 复制协议中的服务器通过蓝色箭头连接。

### 拓扑图示例：推荐的拓扑

以下推荐的拓扑示例显示了四个服务器的可能的推荐拓扑之一：每个服务器至少连接到两个其他服务器，并且多个服务器是一个 CA 服务器。

图 58.4. 推荐的拓扑示例

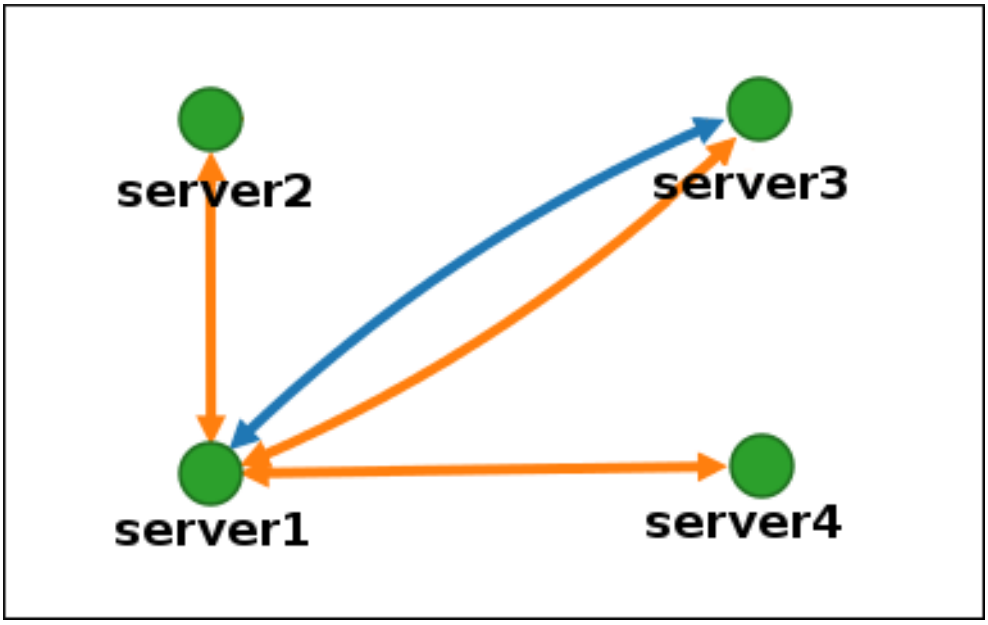



### 拓扑图示例：不建议拓扑

在以下不建议的拓扑示例中，server1 是一个单点故障。所有其他服务器与此服务器具有复制协议，但任何其他服务器都不具有复制协议。因此，如果 server1 出现故障，所有其他服务器将被隔离。

避免创建如下拓扑：

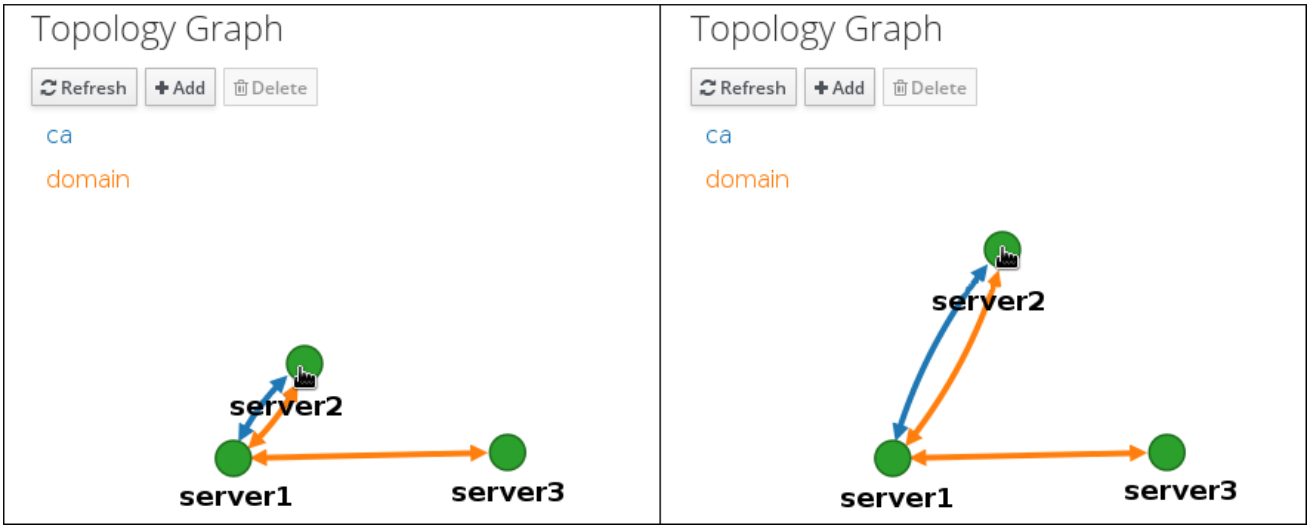
图 58.5. 不推荐的拓扑示例： 单点故障




自定义拓扑视图

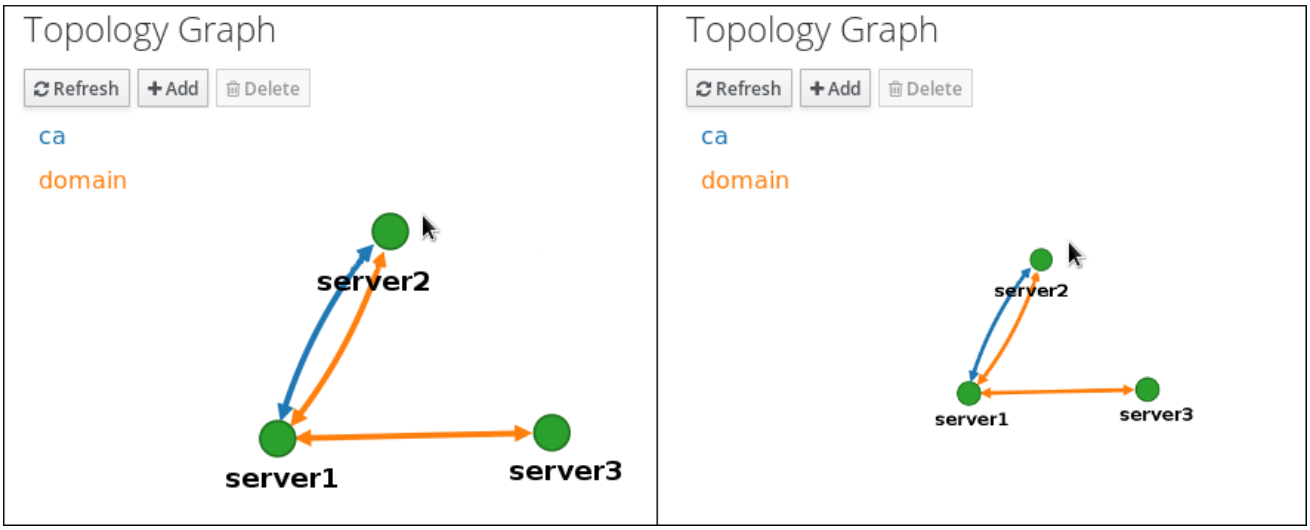
您可以通过拖动鼠标来移动单独的拓扑节点：

图 58.6. 移动拓扑图形节点



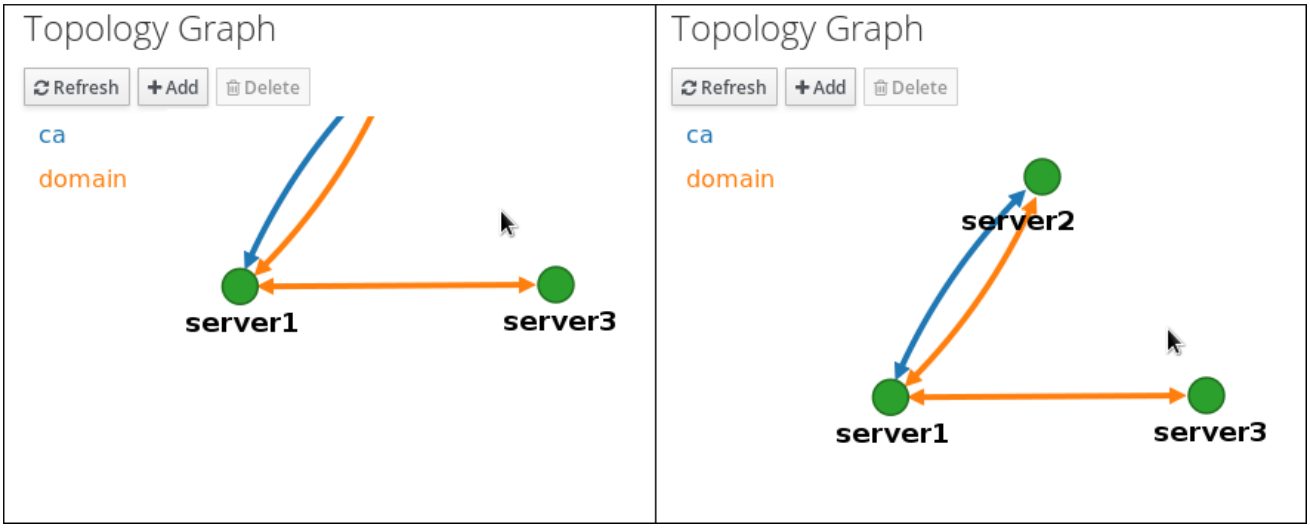

您可以使用鼠标 wheel 缩放拓扑图：

图 58.7. 缩放拓扑图




您可以通过按鼠标左键移动拓扑图的画面：

图 58.8. 移动拓扑图画面




## 58.3. 使用 WEB UI 在两个服务器之间设置复制

利用身份管理(IdM)的 Web 界面，您可以选择两个服务器并在它们之间创建新的复制协议。

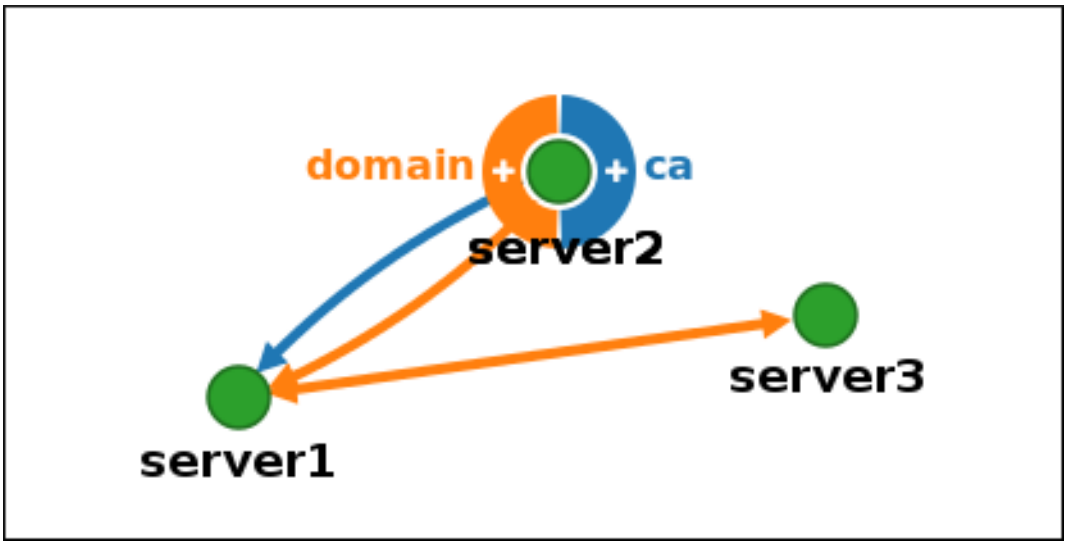
**先决条件**

- 有 IdM 管理员凭证。

**流程**

1. 在拓扑图中，将鼠标悬停在其中一个服务器节点上。

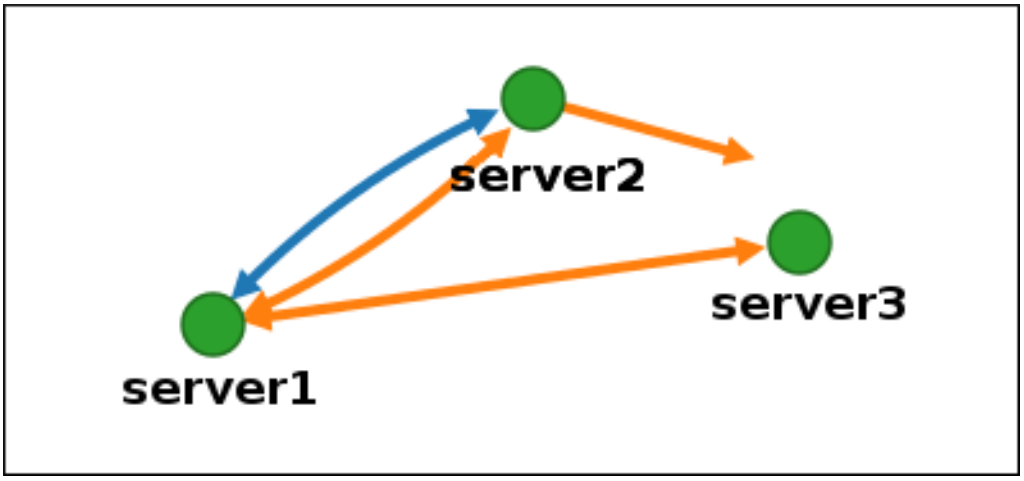
图 58.9. 域或 CA 选项




2. 点击 域 或圆圈的 ca 部分，具体取决于您要创建的拓扑网类型。

3. 在鼠标指针下会显示代表新复制协议的新箭头。将鼠标移到其他服务器节点，然后单击它。

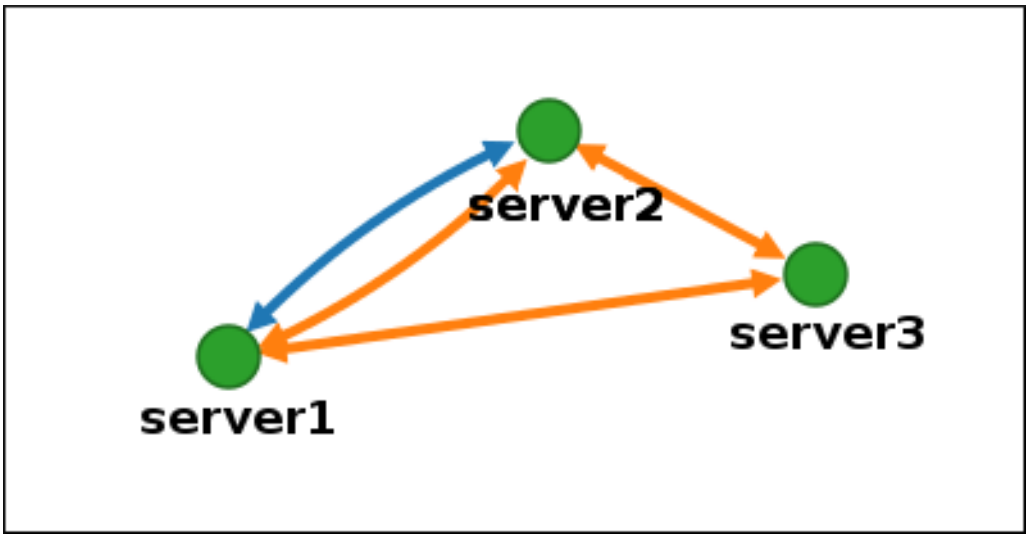
图 58.10. 创建新片段




4. 在 Add topology segment 窗口中，单击 Add 以确认新网段的属性。

两台服务器之间的新拓扑网段将它们加入到复制协议中。拓扑图现在显示更新的复制拓扑：

图 58.11. 创建新片段




## 58.4. 使用 WEB UI 停止两个服务器之间的复制

利用身份管理(IdM)的 Web 界面，您可以从服务器中删除复制协议。

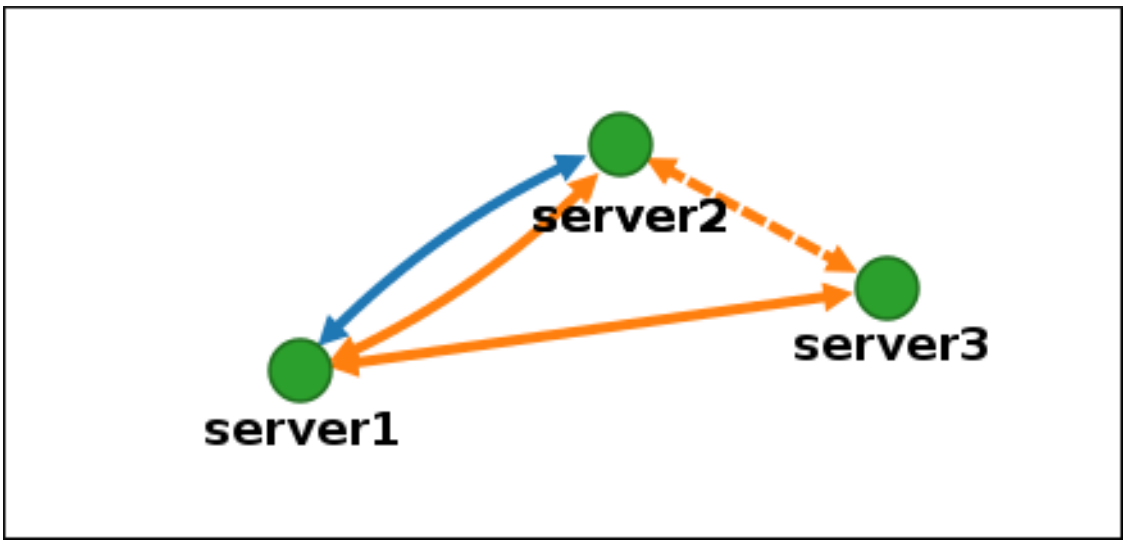
**先决条件**

- 有 IdM 管理员凭证。

**流程**

1. 单击代表您要删除的复制协议的箭头。这会突出显示箭头。

图 58.12. 拓扑片段突出显示

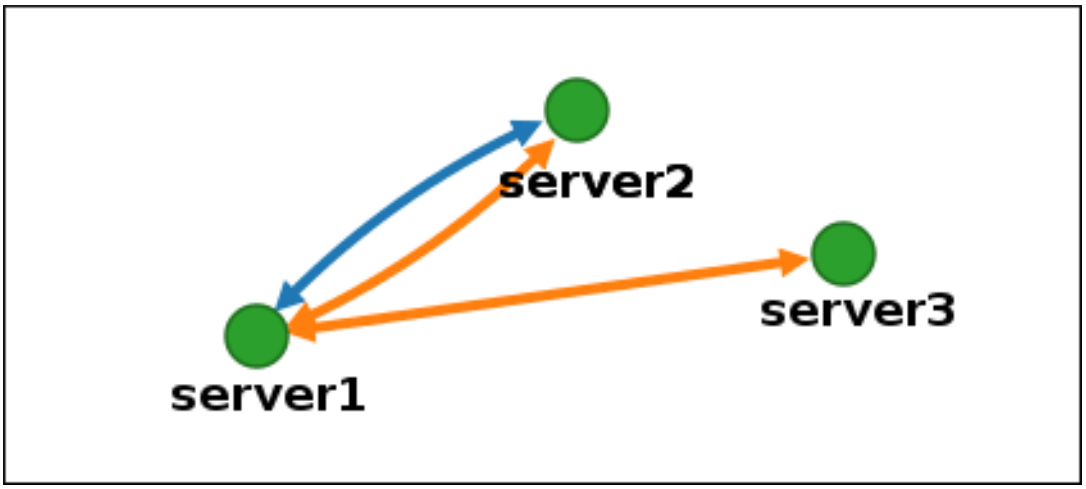


2. 点删除。

3. 在 Confirmation 窗口中，单击 OK。

IdM 删除了两台服务器之间的拓扑网段，这将删除它们的复制协议。拓扑图现在显示更新的复制拓扑：

图 58.13. 已删除拓扑片段




## 58.5. 使用 CLI 在两个服务器之间设置复制

您可以使用 ipa topologysegment-add 命令配置两个服务器之间的复制协议。

**先决条件**

- 有 IdM 管理员凭证。

**流程**

1. 使用 ipa topologysegment-add 命令创建两个服务器的拓扑网段。在提示时，提供：
   - 所需的拓扑后缀： domain 或 ca
   - 左侧节点和右侧节点，代表两台服务器

- （可选）部分的自定义名称

例如：

```
$ ipa topologysegment-add
Suffix name: domain
Left node: server1.example.com
Right node: server2.example.com
Segment name [server1.example.com-to-server2.example.com]: new_segment
----------------------------
Added segment "new_segment"
----------------------------
  Segment name: new_segment
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both
```

添加新片段在复制协议中加入服务器。

2. *可选。*使用 ipa topologysegment-show 命令验证是否已配置新网段。

```
$ ipa topologysegment-show
Suffix name: domain
Segment name: new_segment
  Segment name: new_segment
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both
```

### 58.6. 使用 CLI 停止两个服务器之间的复制

您可以使用 ipa topology segment-del 命令从命令行终止复制协议。

**先决条件**

- 有 IdM 管理员凭证。

**流程**

1. 要停止复制，您必须删除服务器之间的对应复制网段。为此，您需要知道网段名称。

如果您不知道名称，请使用 ipa topologysegment-find 命令显示所有片段，并在输出中找到所需的片段。出现提示时，请提供所需的拓扑后缀： domain 或 ca。例如：

```
$ ipa topologysegment-find
Suffix name: domain
------------------
8 segments matched
------------------
  Segment name: new_segment
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both

...

----------------------------
Number of entries returned 8
----------------------------
```

2. 使用 ipa topologysegment-del 命令删除加入两台服务器的拓扑网段。

```
$ ipa topologysegment-del
Suffix name: domain
Segment name: new_segment
-----------------------------
Deleted segment "new_segment"
-----------------------------
```

删除片段会移除复制协议。

3. *可选。*使用 ipa topologysegment-find 命令验证网段是否不再列出。

```
$ ipa topologysegment-find
Suffix name: domain
------------------
7 segments matched
------------------
  Segment name: server2.example.com-to-server3.example.com
  Left node: server2.example.com
  Right node: server3.example.com
  Connectivity: both

...

----------------------------
Number of entries returned 7
----------------------------
```

## 58.7. 使用 WEB UI 从拓扑中删除服务器

您可以使用 Identity Management(IdM)web 界面从拓扑中删除服务器。

### 先决条件

- 有 IdM 管理员凭证。
- 您希望删除的服务器 并不是 唯一将其他服务器与拓扑中其余部分连接的服务器；这会导致其他服务器变为隔离状态，这是不允许的。
- 要删除的服务器 不是 您的最后一个 CA 或 DNS 服务器。

**警告**

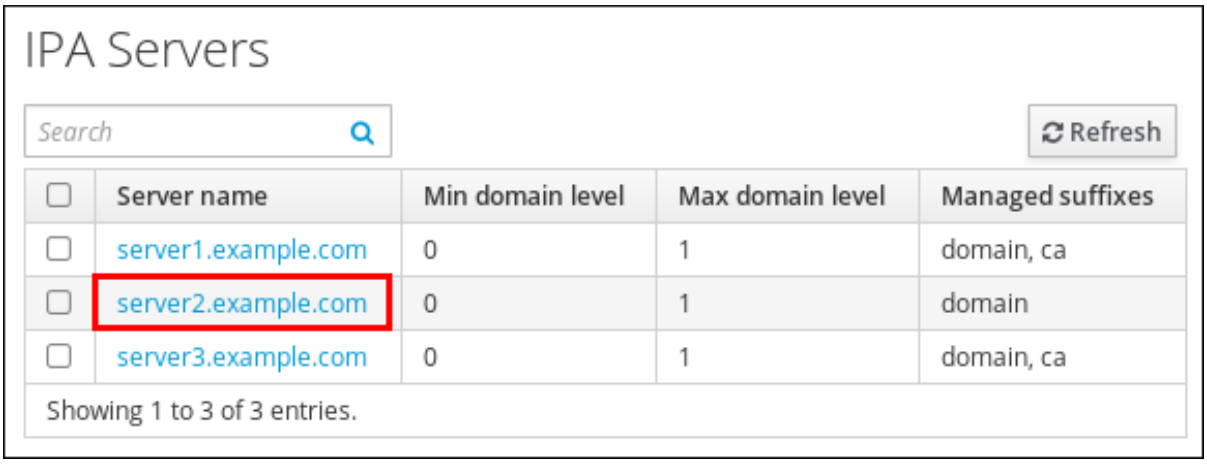
删除服务器是一项不当操作。如果您删除了服务器，在拓扑中重新引入它的唯一方法是在机器上安装新副本。

### 流程

在不从机器中卸载服务器组件的情况下从拓扑中删除服务器：

1. 选择 IPA 服务器 → 拓扑 → IPA 服务器。
2. 单击您要删除的服务器的名称。

图 58.14. 选择服务器




3. 单击 **Delete Server**。

## 58.8. 使用 CLI 从拓扑中删除服务器

您可以使用命令行界面从拓扑中删除服务器。

**先决条件**

- 有 IdM 管理员凭证。
- 您希望删除的服务器 不是 将其他服务器与拓扑其余部分连接的唯一服务器；这会导致其他服务器被隔离，这是不允许的。
- 要删除的服务器 不是 您的最后一个 CA 或 DNS 服务器。

**重要**

删除服务器是一项不当操作。如果您删除了服务器，在拓扑中重新引入它的唯一方法是在机器上安装新副本。

**流程**

删除 server1.example.com ：

1. 在另一台服务器上，运行 ipa server-del 命令来移除 server1.example.com。该命令删除指

向服务器的所有拓扑片段：

```
[user@server2 ~]$ ipa server-del
Server name: server1.example.com
Removing server1.example.com from replication topology, please wait...
----------------------------------------------------------
Deleted IPA server "server1.example.com"
----------------------------------------------------------
```

2. *可选* ：在 server1.example.com 中，运行 ipa server-install --uninstall 命令，从机器中卸载服务器组件。

```
[root@server1 ~]# ipa server-install --uninstall
```

## 58.9. 使用 WEB UI 查看 IDM 服务器上的服务器角色

根据 IdM 服务器中安装的服务，它可以执行各种 *服务器角色*。例如：

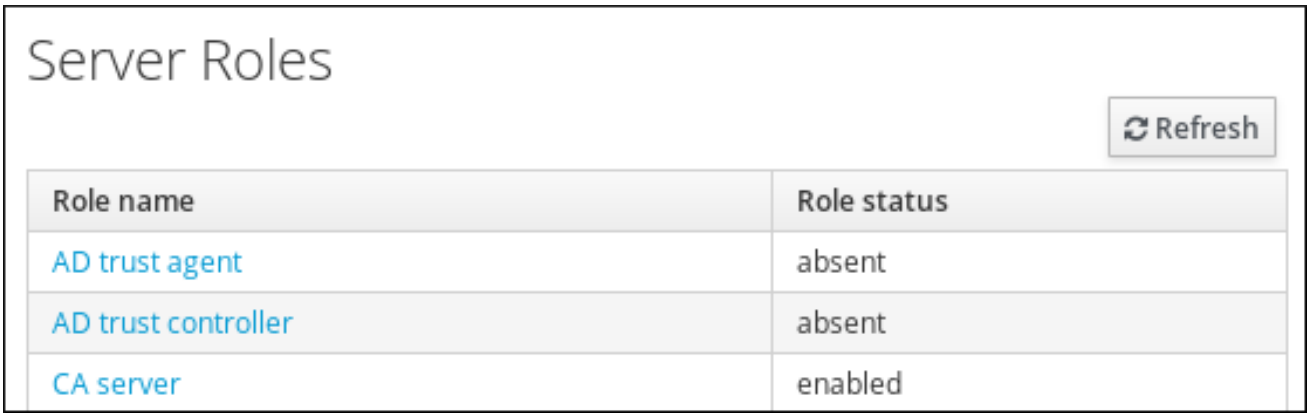
- CA 服务器
- DNS 服务器
- 关键恢复机构(KRA)服务器.

有关支持的服务器角色的完整列表，请参阅 IPA 服务器 → Topology → Server Roles。

注意

- 缺少 角色状态意味着拓扑中没有服务器执行该角色。
- 启用 角色状态意味着拓扑中的一个或多个服务器正在执行该角色。

图 58.15. Web UI 中的服务器角色




## 58.10. 使用 CLI 查看 IDM 服务器上的服务器角色

根据 IdM 服务器中安装的服务，它可以执行各种 *服务器角色*。例如：

- CA 服务器
- DNS 服务器
- 关键恢复机构(KRA)服务器.

您可以使用以下命令查看拓扑中执行哪些服务器：

- ipa config-show 命令显示所有 CA 服务器和当前 CA 续订服务器：

```
$ ipa config-show
  ...
  IPA masters: server1.example.com, server2.example.com, server3.example.com
  IPA CA servers: server1.example.com, server2.example.com
  IPA CA renewal master: server1.example.com
```

- ipa server-show 命令显示在特定服务器上启用的角色列表。例如，列出在 *server.example.com* 中启用的角色列表：

```
$ ipa server-show
Server name: server.example.com
  ...
  Enabled server roles: CA server, DNS server, KRA server
```

- ipa server-find --servrole 搜索启用了特定服务器角色的所有服务器。例如，搜索所有 CA 服务器：

```
$ ipa server-find --servrole "CA server"
---------------------
2 IPA servers matched
---------------------
  Server name: server1.example.com
  ...

  Server name: server2.example.com
  ...
----------------------------
Number of entries returned 2
----------------------------
```

### 58.11. 将副本提升到 CA 续订服务器和 CRL 发布程序服务器

如果您的 IdM 部署使用嵌入式证书颁发机构(CA)，其中一个 IdM CA 服务器充当 CA 续订服务器，负责管理 CA 子系统证书的续订服务器。其中一台 IdM CA 服务器也充当 IdM CRL 发布程序服务器，这是生成证书撤销列表的服务器。默认情况下，CA 续订服务器和 CRL 发布程序服务器角色安装在第一个服务器上，系统管理员使用 ipa-server-install 或 ipa-ca-install 命令在其上安装 CA 角色。

**先决条件**

- 有 IdM 管理员凭证。

**流程**

- 更改当前的 CA 续订服务器。
- 配置副本以生成 CRL。

### 58.12. 演示或提升隐藏副本

安装副本后，您可以配置副本是隐藏还是可见。

有关隐藏副本的详情，请参阅 隐藏副本模式。

如果副本是 CA 续订服务器，请在隐藏此副本前将服务移到另一个副本。

详情请参阅 更改和重置 IdM CA 续订服务器。

流程

- 要隐藏副本，请输入：

```
# ipa server-state replica.idm.example.com --state=hidden
```

另外，您可以使用以下命令使副本可见：

```
# ipa server-state replica.idm.example.com --state=enabled
```

要查看拓扑中所有隐藏的副本的列表，请输入：

```
# ipa config-show
```

如果启用了所有副本，命令输出不会提到隐藏的副本

# 第 59 章 身份管理中的公钥证书

X.509 公钥证书用于验证身份管理(IdM)中的用户、主机和服务。除了身份验证外，X.509 证书还支持数字签名和加密，来提供隐私性、完整性和不可否认性。

证书包含以下信息：

- 证书验证的主题。
- 签发者，即签署证书的 CA。
- 证书有效性的开始和结束日期。
- 证书的有效使用。
- 主题的公钥。

由公钥加密的消息只能由相应的私钥解密。虽然包含的证书和公钥可以公开发布，但用户、主机或服务必须对其私钥保密。

## 59.1. IDM 中的证书颁发机构

证书颁发机构在信任层次结构中操作。在带有内部证书颁发机构(CA)的 IdM 环境中，所有 IdM 主机、用户和服务信任由 CA 签名的证书。除了这个根 CA 外，IdM 还支持根 CA 授予其依次签署证书能力的子 CA。通常，此类子 CA 能够签名的证书是特定类型的证书，如 VPN 证书。最后，IdM 支持使用外部 CA。下 表显示了 在 IdM 中使用独立 CA 的详情。

表 59.1. 在 IdM 中使用集成和外部 CA 的比较

| CA 的名称 | 描述 | 使用 | 有用的链接 |
| --- | --- | --- | --- |
| **ipa** CA | 基于 Dogtag 上游项目的集成 CA | 集成的 CA 可以为用户、主机和服务创建、吊销和发布证书。 | 使用 ipa CA 来请求一个新用户证书，并将其导出到客户端 |
| IdM sub-CAs | 从属于 **ipa** CA 的集成 CA | IdM 子 CA 是 **ipa** CA 对其授予了签署证书的 CA。通常，这些证书是特定类型的，如 VPN 证书。 | 将应用程序限制为只信任证书子集 |
| 外部 CA | 外部 CA 是集成 IdM CA 或其子 CA 以外的 CA。 | 使用 IdM 工具，您可以将这些 CA 发布的证书添加到用户、服务或主机，以及删除它们。 | 管理 IdM 用户、主机和服务的外部签名证书 |

**从证书的角度来看，由自签名 IdM CA 签名和外部签名的证书之间没有区别。**

**CA 的作用包括以下目的：**

- **它发布数字证书。**
- **通过签署证书，它证明证书中指定的对象拥有一个公钥。主题可以是用户、主机或服务。**
- **它可以吊销证书，并通过证书吊销列表(CRL)和在线证书状态协议(OCSP)提供吊销状态。**

**其它资源**

- 请参阅 **规划您的 CA 服务**。

## 59.2. 证书和 KERBEROS 的比较

证书与 **Kerberos 票据执行类似的功能。Kerberos 是一种计算机网络身份验证协议，它在票据的基础上工作，来允许节点通过非安全网络进行通信，从而以安全的方式证明它们相互的身份。下表显示了 Kerberos 和 X.509 证书的比较：**

表 59.2. 证书和 Kerberos 的比较

| 特性 | Kerberos | X.509 |
| --- | --- | --- |
| **Authentication** | 是 | 是 |
| **隐私性** | 可选 | 是 |
| **完整性** | 可选 | 是 |
| **涉及的加密类型** | 对称 | 非对称 |
| **默认有效期** | 短（1 天） | 长（2 年） |

默认情况下，身份管理中的 Kerberos 仅确保通信各方的身份。

## 59.3. 使用证书验证 IDM 中用户的优缺点

在 IdM 中使用证书验证用户的优点包括以下几点：

- 与常规密码相比，智能卡上保护私钥的 PIN 通常不复杂、更容易记住。
- 根据设备的不同，无法导出保存在智能卡上的私钥。这提供了额外的安全性。
- 智能卡可以自动退出登录： IdM 可以配置为在用户从读卡器中移除智能卡时退出用户登录。
- 窃取私钥需要对智能卡的实际访问，这样可以防止智能卡遭受攻击。
- 智能卡验证是一双因素验证的一个示例：它要求您拥有某些东西（卡），知道某些东西 (PIN)。
- 智能卡比密码更灵活，因为它们提供可用于其他用途的密钥，如加密电子邮件。
- 在作为 IdM 客户端的共享机器上使用智能卡不会给系统管理员带来额外的配置问题。事实上，智能卡验证对于共享机器来说是一个理想的选择。

在 IdM 中使用证书验证用户的缺点包括以下几点：

- 用户可能会丢失或忘记携带其智能卡或证书，并被有效锁住。
- 多次输错 PIN 可能会导致卡被锁住。
- 通常，在请求与某些安全官或批准人授权之间有一个中间步骤。在 IdM 中，安全官或管理员必须运行 ipa cert-request 命令。
- 智能卡和读卡器往往是特定于供应商和驱动程序的：虽然许多读卡器可用于不同的卡，但特定供应商的智能卡可能无法在另一供应商的读卡器或不是为其设计的读卡器类型中工作。
- 证书和智能卡对管理员来说有一个陡峭的学习曲线。

# 第 60 章 转换证书格式以和 IDM 一起工作

这个用户故事描述了如何确保您作为 IdM 系统管理员使用正确的带有特定 IdM 命令的证书的格式。例如，这在以下情况下非常有用：

- 您将外部证书加载到用户配置文件中。详情请参阅 转换外部证书以加载到 IdM 用户帐户中 。
- 在为智能卡验证配置 IdM 服务器 或 为智能卡验证配置 IdM 客户端时 ，您在使用外部 CA 证书，以便用户可以使用其上带有由外部证书颁发机构签发的证书的智能卡向 IdM 进行身份验证。
- 您从 NSS 数据库将证书导出为 pkcs #12 格式，其中包括证书和私钥。详情请查看 将 NSS 数据库中的证书和私钥导出到 PKCS #12 文件中。

## 60.1. IDM 中的证书格式和编码

包括 IdM 中的智能卡身份验证的证书验证通过比较用户提供的证书或证书数据（保存在用户的 IdM 配置文件中）来进行。

### 系统配置

IdM 配置文件中存储的内容只是证书，而不是相应的私钥。在身份验证期间，用户还必须显示其拥有相应的私钥。用户通过显示包含证书和私钥的 PKCS #12 文件，或提供两个文件：一个包含证书，另一个包含私钥，来执行此操作。

因此，将证书加载到用户配置文件的进程等只接受不包含私钥的证书文件。

同样，当系统管理员为您提供外部 CA 证书时，他将仅提供公共数据：不带私钥的证书。为 IdM 客户端的智能卡验证配置 IdM 服务器的 ipa-advise 工具需要输入文件包含外部 CA 的证书，而不是私钥。

### 证书编码

有两种常见的证书编码：隐私增强的电子邮件(PEM)和区分的编码规则(DER)。base64 格式与 PEM 格式几乎一样，但它不包含 -----BEGIN CERTIFICATE-----/-----END CERTIFICATE----- 标头和页脚。

已使用 DER 编码的证书是二进制 X509 数字证书文件。作为二进制文件，证书不可读。DER 文件有时使用 .der 文件扩展名，但带有 .crt 和 .cer 文件扩展名的文件有时也会包含 DER 证书。包含密钥的 DER

文件可以命名为 .key。

使用 PEM Base64 编码的证书是一个人类可读的文件。该文件包含前缀为"-----BEGIN ..."的 ASCII(Base64)保护的数据行。PEM 文件有时使用 .pem 文件扩展名，但带有 .crt 和 .cer 文件扩展名的文件有时也包含 PEM 证书。包含密钥的 PEM 文件可以命名为 .key。

不同的 ipa 命令对其接受的证书类型有不同的限制。例如，ipa user-add-cert 命令只接受以 base64 格式编码的证书，但 ipa-server-certinstall 接受 PEM、DER、PKCS #7、PKCS #8 和 PKCS #12 证书。

表 60.1. 证书编码

| 编码格式 | 人类可读 | 常用的文件扩展名 | 接受编码格式的 IdM 命令示例 |
|---|---|---|---|
| PEM/base64 | 是 | .pem, .crt, .cer | ipa user-add-cert, ipa-server-certinstall, ... |
| DER | 否 | .der, .crt, .cer | ipa-server-certinstall, ... |

IdM 中与证书相关的命令和格式 列出了其它 ipa命令以及这些命令可以接受的证书格式。

用户身份验证

在使用 Web UI 访问 IdM 时，用户证明自己通过将两者都存储在浏览器的数据库中，证明自己拥有与证书对应的私钥。

当使用 CLI 访问 IdM 时，用户通过以下方法之一证明自己拥有与证书对应的私钥：

- 用户添加连接到包含证书和密钥的智能卡模块的路径，作为 kinit -X 命令的 X509_user_identity 参数的值：

```
$ kinit -X X509_user_identity='PKCS11:opensc-pkcs11.so' idm_user
```

- 用户添加两个文件作为 kinit -X 命令的 X509_user_identity 参数的值，一个包含证书，另一个包含私钥：

```
$ kinit -X X509_user_identity='FILE:`/path/to/cert.pem,/path/to/cert.key`' idm_user
```

有用的证书命令

查看证书数据，如主题和签发者：

```
$ openssl x509 -noout -text -in ca.pem
```

要比较两个证书在哪些行不同：

```
$ diff cert1.crt cert2.crt
```

要通过两列中显示的输出来比较两个证书在哪些行不同：

```
$ diff cert1.crt cert2.crt -y
```

## 60.2. 将外部证书转换来加载到 IDM 用户帐户中

本节描述了如何确保在将外部证书添加到用户条目之前正确对其进行编码和格式化。

### 60.2.1. 先决条件

- 如果您的证书是由活动目录证书认证机构签发，并使用 PEM 编码的，请确保 PEM 文件已转换为 UNIX 格式。要转换文件，请使用 eponymous 软件包提供的 dos2unix 工具。

### 60.2.2. 在 IdM CLI 中转换外部证书，并将其加载到 IdM 用户帐户中

IdM CLI 只接受 PEM 证书，从中删除了第一行和最后一行（-----BEGIN CERTIFICATE----- 和 -----END CERTIFICATE-----）。

按照以下流程将外部证书转换为 PEM 格式，并使用 IdM CLI 将其添加到 IdM 用户帐户中。

流程

1. 将证书转换为 PEM 格式：

   - 如果您的证书为 DER 格式：

```
$ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

- 如果您的文件为 PKCS #12 格式，其常用文件扩展名为 .pfx 和 .p12，并且包含证书、私钥和其他数据，请使用 openssl pkcs12 工具提取证书。提示时，输入保护存储在文件中的私钥的密码：

```
$ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
Enter Import Password:
```

2. 获取管理员凭证：

```
$ kinit admin
```

3. 使用 IdM CLI 将证书添加到用户帐户中，按照以下方法之一：

- 在将字符串添加到 ipa user-add-cert前，使用 sed 工具删除 PEM 文件的第一行和最后一行（-----BEGIN CERTIFICATE----- 和 -----END CERTIFICATE-----）：

```
$ ipa user-add-cert some_user --certificate="$(sed -e '/BEGIN
CERTIFICATE/d;/END CERTIFICATE/d' cert.pem)"
```

- 将没有第一行和最后一行（-----BEGIN CERTIFICATE----- 和 -----END CERTIFICATE-----）的证书文件的内容复制并粘贴到 ipa user-add-cert 命令中：

```
$ ipa user-add-cert some_user --
certificate=MIIDIzCCAn+gAwIBAgIBATANBgkqhki...
```

**注意**

如果不首先删除第一行和最后一行（-----BEGIN CERTIFICATE----- 和 -----END CERTIFICATE-----），您无法直接将包含证书的 PEM 文件作为输入传给 ipa user-add-cert 命令：

```
$ ipa user-add-cert some_user --cert=some_user_cert.pem
```

此命令会导致产生"ipa: ERROR: Base64 decoding failed: Incorrect padding"错误消息。

4. （可选）检查证书是否被系统接受：

```
[idm_user@r8server]$ ipa user-show some_user
```

### 60.2.3. 在 IdM web UI 中转换外部证书，以便将其加载到 IdM 用户帐户中

按照以下流程将外部证书转换为 PEM 格式，并将其添加到 IdM Web UI 中的 IdM 用户帐户中。

**流程**

1. 使用 CLI，将证书转换为 PEM 格式：

   - 如果您的证书为 DER 格式：

```
$ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

   - 如果您的文件为 PKCS #12 格式，其常用文件扩展名为 .pfx 和 .p12，并且包含证书、私钥和其他数据，请使用 openssl pkcs12 工具提取证书。提示时，输入保护存储在文件中的私钥的密码：

```
$ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
Enter Import Password:
```

2. 在编辑器中打开证书，并复制内容。您可以包含 "-----BEGIN CERTIFICATE-----" 和 "-----END CERTIFICATE-----" 标头和页脚行，但您不必这样做，因为 IdM Web UI 接受 PEM 和 base64 格式。

3. 在 IdM Web UI 中，以安全官身份登录。

4. 前往 Identity → Users → some_user。

5. 单击 Certificates 旁边的 Add。

6. 将证书的 PEM 格式内容粘贴到打开的窗口中。

7. 点 Add。

如果证书被系统接受，您可以在用户配置文件中看到它列在 Certificates 中。

## 60.3. 准备将证书加载到浏览器

在将用户证书导入到浏览器前，请确保证书和相应的私钥为 PKCS #12 格式。通常有两种情况需要额外的准备工作：

- 证书位于 NSS 数据库中。有关在这种情况下如何处理的详情，请参考 将 NSS 数据库中的证书和私钥导出到 PKCS #12 文件中。
- 证书和私钥位于两个单独的 PEM 文件中。有关在这种情况下如何处理的详情，请参考 将证书和私钥 PEM 文件合并到 PKCS #12 文件中。

之后，要将 PEM 格式的 CA 证书以及 PKCS #12 格式的用户证书导入到浏览器中，请按照 配置浏览器以启用证书身份验证 和 以身份管理用户的身份使用证书验证身份管理 Web UI 中的流程。

### 60.3.1. 将证书和私钥从 NSS 数据库导出到 PKCS #12 文件中

**流程**

1. 使用 pk12util 命令将证书从 NSS 数据库导出为 PKCS12 格式。例如，要将昵称为 some_user 的证书从存储在 ~/certdb 目录中的 NSS 数据库导出到 ~/some_user.p12 文件中：

```
$ pk12util -d ~/certdb -o ~/some_user.p12 -n some_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

2. 为 .p12 文件设置合适的权限：

```
# chmod 600 ~/some_user.p12
```

由于 PKCS #12 文件也包含私钥，因此必须对其进行保护，以防止其他用户使用该文件。否则，他们可以模拟用户。

### 60.3.2. 将证书和私钥 PEM 文件合并到 PKCS #12 文件中

按照以下流程将证书和存储在单独的 PEM 文件中的相应密钥合并到 PKCS #12 文件中。

**流程**

- 将存储在 certfile.cer 中的证书和存储在 certfile.key 中的密钥合并到包含证书和密钥的 certfile.p12 文件中：

```
$ openssl pkcs12 -export -in certfile.cer -inkey certfile.key -out certfile.p12
```

## 60.4. IDM 中与证书相关的命令和格式

下表显示了 IdM 中与证书相关的可接受格式的命令。

**表 60.2. IdM 证书命令和格式**

| 命令 | 可接受的格式 | 备注 |
|---|---|---|
| **ipa user-add-cert some_user --certificate** | base64 PEM 证书 | |
| **ipa-server-certinstall** | PEM 和 DER 证书；PKCS#7 证书链；PKCS#8 和原始私钥；PKCS#12 证书和私钥 | |
| **ipa-cacert-manage install** | DER; PEM; PKCS#7 | |
| **ipa-cacert-manage renewal --external-cert-file** | PEM 和 DER 证书; PKCS#7 证书链 | |
| **ipa-ca-install --external-cert-file** | PEM 和 DER 证书; PKCS#7 证书链 | |
| **ipa cert-show <cert serial> --certificate-out /path/to/file.pem** | 不适用 | 创建具有 **<cert_serial>** 序列号证书的 PEM 编码的 **file.pem** 文件。 |
| **ipa cert-show <cert serial> --certificate-out /path/to/file.pem** | 不适用 | 创建具有 **<cert_serial>** 序列号证书的 PEM 编码的 **file.pem** 文件。如果使用 **--chain** 选项，PEM 文件将含有包含证书链的证书。 |

| 命令 | 可接受的格式 | 备注 |
| --- | --- | --- |
| **ipa cert-request --certificate-out=FILE /path/to/req.csr** | 不适用 | 使用新证书创建 PEM 格式的 **req.csr** 文件。 |
| **ipa cert-request --certificate-out=FILE /path/to/req.csr** | 不适用 | 使用新证书创建 PEM 格式的 **req.csr** 文件。如果使用 **--chain** 选项，PEM 文件将含有包含证书链的证书。 |

# 第 61 章 使用集成的 IDM CA 为用户、主机和服务管理证书

要了解更多有关如何使用集成的 CA、ipa CA 及其子 CA 管理身份管理(IdM)中证书的信息，请参阅以下部分：

- 使用 IdM Web UI 为用户、主机或服务请求新证书。
- 使用 IdM CLI 为用户、主机或服务从 IdM CA 请求新证书：
  - 使用 certutil 为用户、主机或服务从 IdM CA 请求新证书
    - 对于使用 certutil 工具从 IdM CA 请求新用户证书，并将其导出到 IdM 客户端的具体示例，请参阅 请求新的用户证书并将其导出到客户端。
  - 使用 openssl 为用户、主机或服务从 IdM CA 请求新证书

您还可以使用 certmonger 工具为来自 IdM CA 的服务请求新证书。如需更多信息，请参阅 使用 certmonger 为来自 IdM CA 的服务请求新证书。

**先决条件**

- 您的 IdM 部署包含一个集成的 CA：
  - 有关如何在 IdM 中规划您的 CA 服务的详情，请参考 规划您的 CA 服务。
  - 有关如何安装带有集成 DNS 和集成 CA 作为根 CA 的 IdM 服务器的详情，请参考 安装 IdM 服务器：带有集成 DNS，带有集成 CA 作为根 CA
  - 有关如何安装带有集成 DNS 和外部 CA 作为根 CA 的 IdM 服务器的详情，请参考 安装 IdM 服务器：带有集成 DNS，带有外部 CA 作为根 CA
  - 有关如何安装没有集成 DNS 且集成 CA 作为根 CA 的 IdM 服务器的详情，请参考 安装 IdM 服务器：没有集成 DNS，集成 CA 作为根 CA。

- [可选] 您的 IdM 部署支持使用证书进行用户身份验证：

  - 有关如何配置 IdM 部署以支持使用存储在 IdM 客户端文件系统中的证书进行用户身份验证的详情，请参考 使用存储在 IdM 客户端桌面上的证书配置身份验证。

  - 有关如何配置 IdM 部署以支持使用存储在插入 IdM 客户端的智能卡中的证书进行用户身份验证的详情，请参考 为智能卡身份验证配置身份管理。

  - 有关如何配置 IdM 部署以支持使用由活动目录证书系统发布的智能卡进行用户身份验证的详情，请参考 为 IdM 中的智能卡身份验证配置由 ADCS 发布的证书。

## 61.1. 使用 IDM WEB UI 为用户、主机或服务请求新证书

按照以下流程，使用身份管理(IdM) Web UI 为集成 IdM 证书颁发机构(CA)中的任何 IdM 实体请求新证书：ipa CA 或其任何子 CA。

IdM 实体包括：

- 用户
- 主机
- 服务

重要

通常运行在存储私钥的专用服务节点上的服务。将服务的私钥复制到 IdM 服务器被视为不安全。因此，在为服务请求证书时，请在服务节点上创建证书签名请求(CSR)。

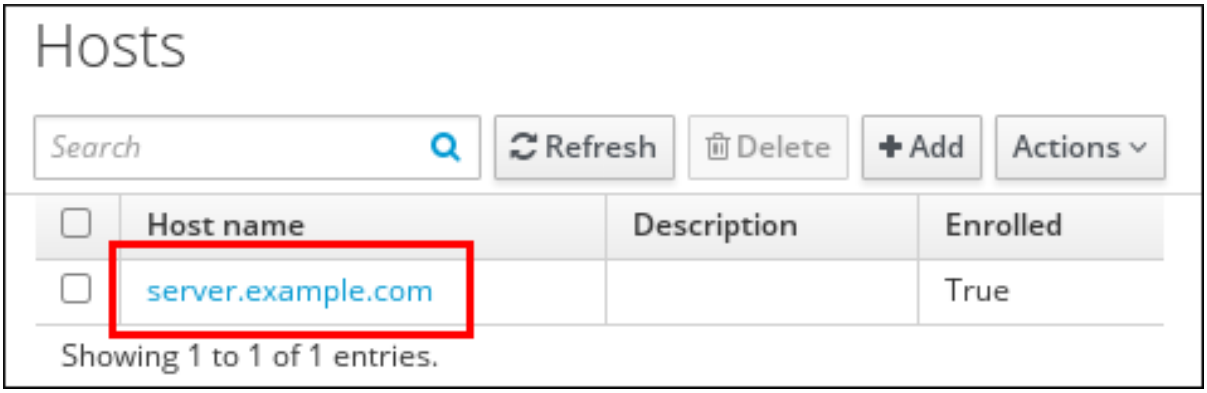
先决条件

- 您的 IdM 部署包含一个集成的 CA。
- 以 IdM 管理员身份登录到 IdM Web UI。

**流程**

1. 在 Identity 选项卡下，选择 Users、Hosts 或 Services 子选项卡。
2. 单击用户、主机或服务的名称，来打开其配置页面。

图 61.1. 主机列表



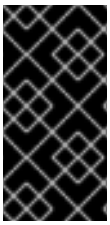

3. 单击 Actions → New Certificate。
4. 可选：选择发布 CA 和配置文件 ID。
5. 按照屏幕上使用 certutil 命令行(CLI)工具的说明进行操作。
6. 单击 Issue。

## 61.2. 使用 CERTUTIL 为用户、主机或服务从 IDM CA 请求新证书

您可以使用 certutil 工具为标准 IdM 情况下的身份管理(IdM)用户、主机或服务请求证书。要确保主机或服务 Kerberos 别名可以使用证书，请 使用 openssl 工具来请求证书。

按照以下流程，使用 certutil 为来自 ipa 、IdM 证书颁发机构(CA)的 IdM 用户、主机或服务请求证

书。



**重要**

通常运行在存储私钥的专用服务节点上的服务。将服务的私钥复制到 IdM 服务器被视为不安全。因此，在为服务请求证书时，请在服务节点上创建证书签名请求(CSR)。

**先决条件**

- 您的 IdM 部署包含一个集成的 CA。
- 以 IdM 管理员身份登录到 IdM 命令行界面(CLI)。

**流程**

1. 为证书数据库创建一个临时目录：

```
# mkdir ~/certdb/
```

2. 创建一个新的临时证书数据库，例如：

```
# certutil -N -d ~/certdb/
```

3. 创建 CSR，并将输出重定向到文件。例如，要为 4096 位证书创建 CSR，并将主题设为 *CN=server.example.com,O=EXAMPLE.COM* ：

```
# certutil -R -d ~/certdb/ -a -g 4096 -s "CN=server.example.com,O=EXAMPLE.COM" -8
server.example.com > certificate_request.csr
```

4. 将证书请求文件提交到在 IdM 服务器上运行的 CA。指定 Kerberos 主体来与新发布的证书关联：

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

IdM 中的 ipa cert-request 命令使用以下默认值：

- caIPAserviceCert 证书配置文件

  要选择自定义配置文件，请使用 --profile-id 选项。

- 集成的 IdM 根 CA ipa

  要选择子 CA，请使用 --ca 选项。

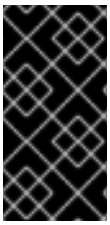
其它资源

- 请参阅 ipa cert-request --help 命令的输出。
- 请参阅 在身份管理中创建和管理证书配置文件。

## 61.3. 使用 OPENSSL 为用户、主机或服务从 IDM CA 请求新证书

如果要确保主机或服务的 Kerberos 别名可以使用证书，您可以使用 openssl 工具为身份管理(IdM)主机或服务请求证书。在标准情况下，请考虑 使用 certutil 工具来请求一个新证书。

按照以下流程，使用 openssl 为 IdM 主机或来自 ipa、IdM 证书颁发机构的服务请求证书。



重要

通常运行在存储私钥的专用服务节点上的服务。将服务的私钥复制到 IdM 服务器被视为不安全。因此，在为服务请求证书时，请在服务节点上创建证书签名请求(CSR)。

先决条件

- 您的 IdM 部署包含一个集成的 CA。
- 以 IdM 管理员身份登录到 IdM 命令行界面(CLI)。

流程

1.

为您的 Kerberos 主体 *test/server.example.com* 创建一个或多个别名。例如，*test1/server.example.com* 和 *test2/server.example.com*。

2. 在 CSR 中，为 dnsName(*server.example.com*)和otherName(*test2/server.example.com*)添加 subjectAltName。要做到这一点，将 openssl.conf 文件配置为包含以下指定 UPN otherName 和 subjectAltName 的行：

```
otherName=1.3.6.1.4.1.311.20.2.3;UTF8:test2/server.example.com@EXAMPLE.COM
DNS.1 = server.example.com
```

3. 使用 openssl 创建证书请求：

```
openssl req -new -newkey rsa:2048 -keyout test2service.key -sha256 -nodes -out
certificate_request.csr -config openssl.conf
```

4. 将证书请求文件提交到在 IdM 服务器上运行的 CA。指定 Kerberos 主体来与新发布的证书关联：

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

IdM 中的 ipa cert-request 命令使用以下默认值：

- caIPAserviceCert 证书配置文件

  要选择自定义配置文件，请使用 --profile-id 选项。

- 集成的 IdM 根 CA ipa

  要选择子 CA，请使用 --ca 选项。

**其它资源**

- 请参阅 ipa cert-request --help 命令的输出。
- 请参阅 在身份管理中创建和管理证书配置文件。

### 61.4. 其它资源

- 请参阅 撤销带有集成 IdM CA 的证书。
- 请参阅 恢复带有集成 IdM CA 的证书。
- 请参阅 将应用程序限制为只信任证书子集。

# 第 62 章 使用 ANSIBLE 管理 IDM 证书

您可以使用 ansible-freeipa ipacert 模块为身份管理(IdM)用户、主机和服务请求、撤销和检索 SSL 证书。您还可以恢复已搁置的证书。

## 62.1. 使用 ANSIBLE 为 IDM 主机、服务和用户请求 SSL 证书

您可以使用 ansible-freeipa ipacert 模块为身份管理(IdM)用户、主机和服务请求 SSL 证书。然后，他们可以使用这些证书向 IdM 进行身份验证。

完成此流程，使用 Ansible playbook ，从 IdM 证书颁发机构(CA)为 HTTP 服务器请求一个证书。

**先决条件**

- 在控制节点上：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已安装 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。
- 您的 IdM 部署有一个集成的 CA。

**流程**

1. 为您的用户、主机或服务生成一个证书签名请求(CSR)。例如，要使用 openssl 工具为运行在 client.idm.example.com 上的 HTTP 服务生成一个 CSR，请输入：

```
# openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout new.key -out new.csr -
subj '/CN=client.idm.example.com,O=IDM.EXAMPLE.COM'
```

因此，CSR 存储在 new.csr 中。

2. 使用以下内容创建您的 Ansible playbook 文件 request-certificate.yml ：

```
---
- name: Playbook to request a certificate
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Request a certificate for a web server
    ipacert:
      ipaadmin_password: "{{ ipaadmin_password }}"
      state: requested
      csr: |
        -----BEGIN CERTIFICATE REQUEST-----

MIGYMEwCAQAwGTEXMBUGA1UEAwwOZnJlZWlwYSBydWxlcyEwKjAFBgMrZXADIQBs
        HlqIr4b/XNK+K8QLJKIzfvuNK0buBhLz3LAzY7QDEqAAMAUGAytlcANBAF4oSCbA
        5aIPukCidnZJdr491G4LBE+URecYXsPknwYb+V+ONnf5ycZHyaFv+jkUBFGFeDgU
        SYaXm/gF8cDYjQI=
        -----END CERTIFICATE REQUEST-----
      principal: HTTP/client.idm.example.com
    register: cert
```

将证书请求替换为 new.csr 中的 CSR。

3. 请求证书：

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/request-
certificate.yml
```

**其它资源**

- ansible-freeipa 上游文档中的 cert 模块

## 62.2. 使用 ANSIBLE 撤销 IDM 主机、服务和用户的 SSL 证书

您可以使用 ansible-freeipa ipacert 模块撤销身份管理(IdM)用户、主机和服务向 IdM 进行身份验证所使用的 SSL 证书。

完成此流程，使用 Ansible playbook 撤销 HTTP 服务器的证书。吊销证书的原因是 "keyCompromise"。

**先决条件**

- 在控制节点上：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已安装 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。
  - 您已获得了证书的序列号，例如输入 openssl x509 -noout -text -in <path_to_certificate> 命令。在本例中，证书的序列号为 123456789。
- 您的 IdM 部署有一个集成的 CA。

**流程**

1. 使用以下内容创建 Ansible playbook 文件 revoke-certificate.yml ：

```
---
- name: Playbook to revoke a certificate
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Revoke a certificate for a web server
    ipacert:
      ipaadmin_password: "{{ ipaadmin_password }}"
```

```
    serial_number: 123456789
    revocation_reason: "keyCompromise"
    state: revoked
```

2. 吊销证书：

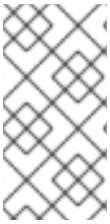
```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/revoke-
certificate.yml
```

其它资源

- ansible-freeipa 上游文档中的 cert 模块
- RFC 5280 中的 原因代码

## 62.3. 使用 ANSIBLE 恢复 IDM 用户、主机和服务的 SSL 证书

您可以使用 ansible-freeipa ipacert 模块恢复之前身份管理(IdM)用户、主机或服务用来向 IdM 进行身份验证时撤销的 SSL 证书。



注意

您只能恢复搁置的证书。您可能已搁置了它，例如，您不确定私钥是否已丢失。但是，您现在已恢复了密钥，并且您确定没有人同时访问它，您希望重新恢复证书。

完成此流程，使用 Ansible playbook 为注册到 IdM 的服务发放搁置的证书。这个示例描述了如何为 HTTP 服务发放搁置的证书。

先决条件

- 在控制节点上：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已安装 ansible-freeipa 软件包。

- 您已在 *~/MyPlaybooks/* 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
- 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。

- 您的 IdM 部署有一个集成的 CA。
- 您已获得了证书的序列号，例如通过输入 openssl x509 -noout -text -in path/to/certificate 命令。在本例中，证书序列号为 123456789。

**流程**

1. 使用以下内容创建 Ansible playbook 文件 restore-certificate.yml ：

```
---
- name: Playbook to restore a certificate
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Restore a certificate for a web service
    ipacert:
      ipaadmin_password: "{{ ipaadmin_password }}"
      serial_number: 123456789
      state: released
```

2. 运行 playbook ：

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/restore-
certificate.yml
```

**其它资源**

- ansible-freeipa 上游文档中的 cert 模块

## 62.4. 使用 ANSIBLE 检索 IDM 用户、主机和服务的 SSL 证书

您可以使用 ansible-freeipa ipacert 模块检索为身份管理(IdM)用户、主机或服务发放的 SSL 证书，并将其存储在受管节点上的一个文件中。

**先决条件**

- 在控制节点上：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已安装 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。
- 您已获得了证书的序列号，例如输入 openssl x509 -noout -text -in <path_to_certificate> 命令。在本例中，证书的序列号为 123456789，存储检索到的证书的文件是 cert.pem。

**流程**

1. 使用以下内容创建 Ansible playbook 文件 retrieve-certificate.yml ：

```
---
- name: Playbook to retrieve a certificate and store it locally on the managed node
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Retrieve a certificate and save it to file 'cert.pem'
    ipacert:
      ipaadmin_password: "{{ ipaadmin_password }}"
      serial_number: 123456789
      certificate_out: cert.pem
      state: retrieved
```

2. 检索证书：

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/retrieve-
certificate.yml
```

**其它资源**

- ansible-freeipa 上游文档中的 cert 模块

# 第 63 章 管理 IDM 用户、主机和服务的外部签名证书

本章描述了如何使用身份管理(IdM)命令行界面(CLI)和 IdM Web UI 来添加或删除用户、主机，以及由外部证书颁发机构(CA)发布的服务证书。

## 63.1. 使用 IDM CLI ，将外部 CA 发布的证书添加到 IDM 用户、主机或服务

作为身份管理(IdM)管理员，您可以使用身份管理(IdM)CLI 向 IdM 用户的帐户、主机或服务添加外部签名的证书。

**先决条件**

- 您已获得管理员用户的票据授予票据。

**流程**

- 要为 IdM 用户添加证书，请输入：

```
$ ipa user-add-cert user --certificate=MIQTPrajQAwg...
```

该命令要求您指定以下信息：

  - 用户名
  - Base64 编码的 DER 证书

**注意**

不是将证书内容复制并粘贴到命令行，您可以将证书转换为 DER 格式，然后将其重新编码为 Base64。例如，要将 user_cert.pem 证书添加给 user，请输入：

```
$ ipa user-add-cert user --certificate="$(openssl x509 -outform der -in
user_cert.pem | base64 -w 0)"
```

您可以在不添加任何选项的情况下，以交互方式运行 ipa user-add-cert 命令。

要将证书添加给 IdM 主机，请输入：

- ipa host-add-cert

要将证书添加给 IdM 服务，请输入：

- ipa service-add-cert

**其它资源**

- 使用集成的 IdM CA 为用户、主机和服务管理证书

## 63.2. 使用 IDM WEB UI 将外部 CA 发布的证书添加到 IDM 用户、主机或服务中

作为身份管理(IdM)管理员，您可以使用身份管理(IdM)Web UI 将外部签名的证书添加到 IdM 用户的帐户、主机或服务中。

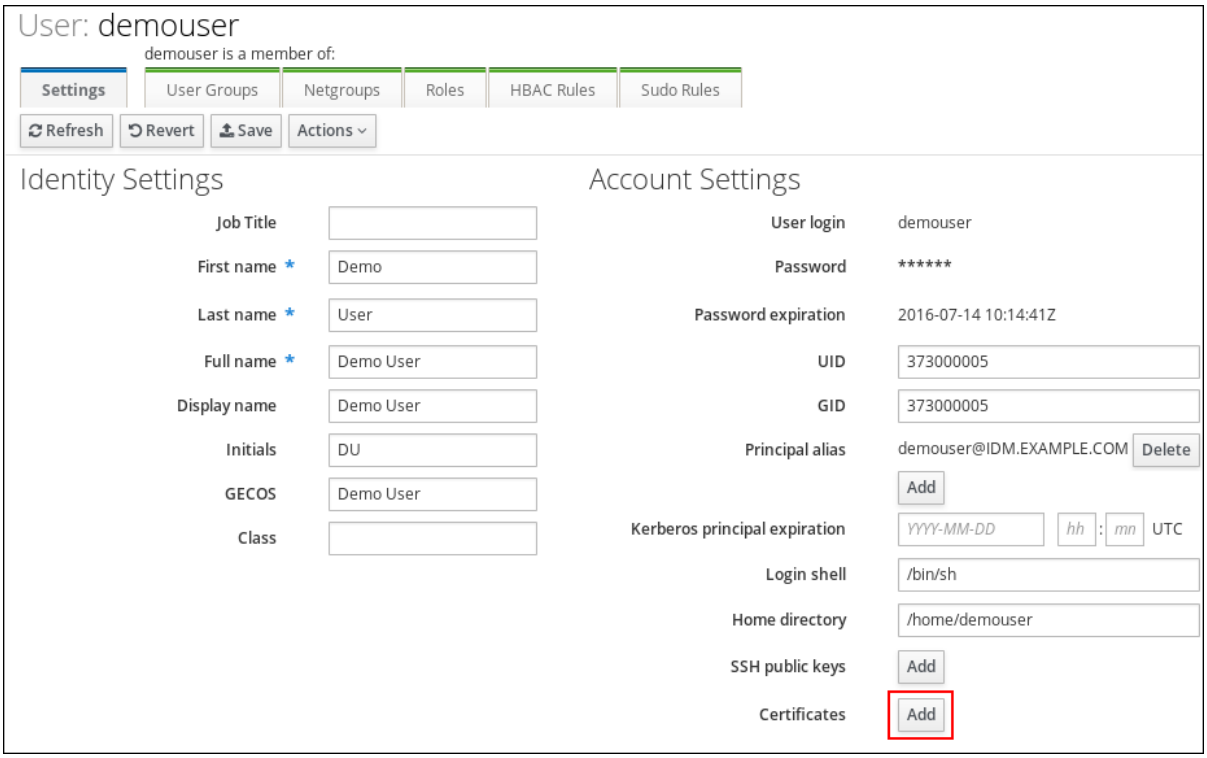
**先决条件**

- 您以管理用户的身份登录到身份管理(IdM)Web UI。

**流程**

1. 打开 Identity 选项卡，然后选择 Users、Hosts 或 Services 子选项卡。
2. 单击用户、主机或服务的名称，来打开其配置页面。
3. 单击 Certificates 条目旁边的 Add。

图 63.1. 将证书添加给用户帐户




4. 将 Base64 或 PEM 编码格式的证书粘贴到文本字段中，然后单击 Add。

5. 单击 Save 以保存更改。

## 63.3. 使用 IDM CLI 从 IDM 用户、主机或服务帐户中删除外部 CA 发布的证书

作为身份管理(IdM)管理员，您可以使用身份管理(IdM)CLI 从 IdM 用户的帐户、主机或服务中删除外部签名的证书。

### 先决条件

- 您已获得管理员用户的票据授予票据。

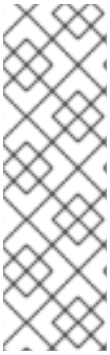
### 流程

- 要从 IdM 用户中删除证书，请输入：

```
$ ipa user-remove-cert user --certificate=MIQTPrajQAwg...
```

该命令要求您指定以下信息：

- 用户名
- Base64 编码的 DER 证书



**注意**

不是将证书内容复制并粘贴到命令行，您可以将证书转换为 DER 格式，然后将其重新编码为 Base64。例如，要从user 中删除 user_cert.pem 证书，请输入：

```
$ ipa user-remove-cert user --certificate="$(openssl x509 -outform der -in
user_cert.pem | base64 -w 0)"
```

您可以在不添加任何选项的情况下，以交互方式运行 ipa user-remove-cert 命令。

要从 IdM 主机中删除证书，请输入：

- ipa host-remove-cert

要从 IdM 服务中删除证书，请输入：

- ipa service-remove-cert

其它资源

- 使用集成的 IdM CA 为用户、主机和服务管理证书

## 63.4. 使用 IDM WEB UI 从 IDM 用户、主机或服务帐户中删除外部 CA 发布的证书

作为身份管理(IdM)管理员，您可以使用身份管理(IdM)Web UI 从 IdM 用户的帐户、主机或服务中删除外部签名的证书。

**先决条件**

- 您以管理用户的身份登录到身份管理(IdM)Web UI。

**流程**

1. 打开 Identity 选项卡，然后选择 Users、Hosts 或 Services 子选项卡。
2. 单击用户、主机或服务的名称，来打开其配置页面。
3. 单击要删除的证书旁边的 Actions，然后选择 Delete。
4. 单击 Save 以保存更改。

### 63.5. 其它资源

- 使用 Ansible playbook 确保 IdM 服务条目中存在外部签名的证书

# 第 64 章 在身份管理中创建和管理证书配置文件

证书授权机构(CA)在签名证书时使用证书配置文件，来确定证书签名请求(CSR)是否可以接受，如果可以接受，证书上有哪些功能和扩展。证书配置文件与发布特定类型的证书相关联。通过组合证书配置文件和 CA 访问控制列表(ACL)，您可以定义和控制对自定义证书配置文件的访问。

在描述如何创建证书配置集时，流程使用 S/MIME 证书作为示例。某些电子邮件程序支持使用安全多用途互联网邮件扩展(S/MIME)协议进行数字签名和加密的电子邮件。使用 S/MIME 签名或加密电子邮件消息，要求消息的发送方具有 S/MIME 证书。

- 什么是证书配置文件
- 创建证书配置文件
- 什么是 CA 访问控制列表
- 定义 CA ACL 来控制对证书配置文件的访问
- 使用证书配置文件和 CA ACL 来发布证书
- 修改证书配置文件
- 证书配置文件配置参数

## 64.1. 什么是证书配置文件？

您可以使用证书配置文件来确定证书的内容，以及发布证书的限制，如下所示：

- 用于隔离证书签名请求的签名算法。
- 证书的默认有效期。

- 用于吊销证书的吊销原因。
- 如果主体的通用名称被复制到主题备用名称字段。
- 应该出现在证书中的功能和扩展。

单个证书配置文件与签发特定类型的证书相关联。您可以在 IdM 中为用户、服务和主机定义不同的证书配置文件。IdM 默认包括以下证书配置文件：

- caIPAserviceCert
- IECUserRoles
- KDCs_PKINIT_Certs （内部使用）

另外，您可以创建和导入自定义配置文件，其允许您为特定目的发布证书。例如，您可以将特定配置文件的使用限制给一个用户或一个组，防止其他用户和组使用该配置文件发布用于身份验证的证书。要创建自定义证书配置文件，请使用 ipa certprofile 命令。

**其它资源**

- 请参阅 ipa help certprofile 命令。

### 64.2. 创建证书配置文件

按照以下流程，通过为请求 S/MIME 证书创建一个证书配置文件，通过命令行来创建一个配置文件。

**流程**

1. 通过复制现有的默认配置文件来创建自定义配置文件：

```
$ ipa certprofile-show --out smime.cfg caIPAserviceCert
----------------------------------------------
```

```
Profile configuration stored in file 'smime.cfg'
----------------------------------------------
  Profile ID: calPAserviceCert
  Profile description: Standard profile for network services
  Store issued certificates: TRUE
```

2. 在文本编辑器中打开新创建的配置文件。

```
$ vi  smime.cfg
```

3. 将 Profile ID 更改为反映配置文件用法的名称，如 smime。

**注意**

当您导入新创建的配置文件时，如果有profileId 字段，则其必须与命令行中指定的 ID 匹配。

4. 更新扩展的密钥用法配置。默认的扩展的密钥用法扩展配置用于 TLS 服务器和客户端身份验证。例如，对于 S/MIME，必须为电子邮件保护配置扩展的密钥用法：

```
policyset.serverCertSet.7.default.params.exKeyUsageOIDs=1.3.6.1.5.5.7.3.4
```

5. 导入新配置文件：

```
$ ipa certprofile-import smime --file smime.cfg \
  --desc "S/MIME certificates" --store TRUE

-----------------------
Imported profile "smime"
-----------------------
  Profile ID: smime
  Profile description: S/MIME certificates
  Store issued certificates: TRUE
```

验证步骤

- 验证新证书配置文件已被导入：

```
$ ipa certprofile-find

-----------------
```

```
4 profiles matched
-----------------
  Profile ID: caIPAserviceCert
  Profile description: Standard profile for network services
  Store issued certificates: TRUE

  Profile ID: IECUserRoles
  Profile description: User profile that includes IECUserRoles extension from request
  Store issued certificates: TRUE

  Profile ID: KDCs_PKINIT_Certs
  Profile description: Profile for PKINIT support by KDCs
  Store issued certificates: TRUE

  Profile ID: smime
  Profile description: S/MIME certificates
  Store issued certificates: TRUE
----------------------------
Number of entries returned 4
----------------------------
```

**其它资源**

- 请参阅 **ipa help certprofile**。
- 请参阅 **RFC 5280 ， 4.2.1.12 部分**。

### 64.3. 什么是 CA 访问控制列表？

证书颁发机构访问控制列表(CA ACL)规则定义哪些配置文件可用于向哪些主体发布证书。您可以使用 CA ACL 来执行此操作，例如：

- 确定可以使用特定配置文件向哪些用户、主机或服务发布证书
- 确定允许哪个 IdM 证书颁发机构或子 CA 发布证书

例如，使用 CA ACL ， 您可以将只用于伦敦办事处工作的员工的配置文件限制为与伦敦办事处相关的 IdM 用户组的成员。

用于管理 CA ACL 规则的 ipa caacl 工具允许特权用户添加、显示、修改或删除指定的 CA ACL。

**其它资源**

**其它资源**

- 请参阅 **ipa help caacl**。

## 64.4. 定义 CA ACL 来控制对证书配置文件的访问

按照以下流程，使用 **caacl** 工具定义一个 CA 访问控制列表(ACL)规则，以允许组中的用户访问自定义证书配置文件。在这种情况下，流程描述了如何创建 S/MIME 用户的组以及 CA ACL，以允许该组中的用户访问 **smime** 证书配置文件。

**先决条件**

- 确保您已获取 IdM 管理员的凭据。

**流程**

1. 为证书配置文件的用户创建一个新组：

```
$ ipa group-add smime_users_group
--------------------------------
Added group "smime users group"
--------------------------------
  Group name: smime_users_group
  GID: 75400001
```

2. 创建一个新用户来添加到 **smime_user_group** 组中：

```
$ ipa user-add smime_user
First name: smime
Last name: user
----------------------
Added user "smime_user"
----------------------
  User login: smime_user
  First name: smime
  Last name: user
  Full name: smime user
  Display name: smime user
  Initials: TU
  Home directory: /home/smime_user
  GECOS: smime user
  Login shell: /bin/sh
  Principal name: smime_user@IDM.EXAMPLE.COM
  Principal alias: smime_user@IDM.EXAMPLE.COM
  Email address: smime_user@idm.example.com
```

```
  UID: 1505000004
  GID: 1505000004
  Password: False
  Member of groups: ipausers
  Kerberos keys available: False
```

3. 将 smime_user 添加到 smime_users_group 组中：

```
$ ipa group-add-member smime_users_group --users=smime_user
  Group name: smime_users_group
  GID: 1505000003
  Member users: smime_user
-------------------------
Number of members added 1
-------------------------
```

4. 创建 CA ACL 以允许组中的用户访问证书配置文件：

```
$ ipa caacl-add smime_acl
------------------------
Added CA ACL "smime_acl"
------------------------
  ACL name: smime_acl
  Enabled: TRUE
```

5. 将用户组添加到 CA ACL 中：

```
$ ipa caacl-add-user smime_acl --group smime_users_group
  ACL name: smime_acl
  Enabled: TRUE
  User Groups: smime_users_group
-------------------------
Number of members added 1
-------------------------
```

6. 将证书配置文件添加到 CA ACL 中：

```
$ ipa caacl-add-profile smime_acl --certprofile smime
  ACL name: smime_acl
  Enabled: TRUE
  Profiles: smime
  User Groups: smime_users_group
-------------------------
Number of members added 1
-------------------------
```

验证步骤

- 查看您创建的 CA ACL 的详情：

```
$ ipa caacl-show smime_acl
  ACL name: smime_acl
  Enabled: TRUE
  Profiles: smime
  User Groups: smime_users_group
...
```

其它资源

- 请参阅 ipa 手册页。
- 请参阅 ipa help caacl。

## 64.5. 使用证书配置文件和 CA ACL 来发布证书

当证书颁发机构访问控制列表(CA ACL)允许时，您可以使用证书配置文件来请求证书。按照以下流程，使用已通过 CA ACL 授予了访问权限的自定义证书配置文件来为用户请求 S/MIME 证书。

先决条件

- 您的证书配置文件已创建。
- 允许用户使用所需证书配置文件请求证书的 CA ACL已创建。

注意

您可以绕过 CA ACL 检查用户是否执行了 cert-request 命令：

- 是 admin 用户。
- 具有 请求证书忽略 CA ACL 权限.

**流程**

1. 为用户生成证书请求。例如，使用 OpenSSL：

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout private.key -out
cert.csr -subj '/CN=smime_user'
```

2. 为用户从 IdM CA 请求新证书：

```
$ ipa cert-request cert.csr --principal=smime_user --profile-id=smime
```

（可选）将 **--ca** *sub-CA_name* 选项传给命令，以从子 CA，而不是根 CA 请求证书。

**验证步骤**

- 验证新发布的证书是否已分配给用户：

```
$ ipa user-show user
  User login: user
  ...
  Certificate: MIICfzCCAWcCAQA...
  ...
```

**其它资源**

- 请参阅 **ipa(a)** 手册页。
- 请参阅 **ipa help user-show** 命令。
- 请参阅 **ipa help cert-request** 命令。
- 请参阅 **openssl(lssl)** 手册页。

### 64.6. 修改证书配置文件

按照以下流程，使用 **ipa certprofile-mod** 命令直接通过命令行修改证书配置文件。

**流程**

1. 确定您要修改的证书配置文件的证书配置文件 ID。显示当前存储在 IdM 中的所有证书配置文件：

```
# ipa certprofile-find

-----------------
4 profiles matched
-----------------
  Profile ID: caIPAserviceCert
  Profile description: Standard profile for network services
  Store issued certificates: TRUE

  Profile ID: IECUserRoles
  ...

  Profile ID: smime
  Profile description: S/MIME certificates
  Store issued certificates: TRUE
----------------------------
Number of entries returned
----------------------------
```

2. 修改证书配置文件描述。例如，如果您使用现有的配置文件为 S/MIME 证书创建了自定义证书配置文件，请按照新用法更改描述：

```
# ipa certprofile-mod smime --desc "New certificate profile description"
-----------------------------------
Modified Certificate Profile "smime"
-----------------------------------
   Profile ID: smime
   Profile description: New certificate profile description
   Store issued certificates: TRUE
```

3. 在文本编辑器中打开您的客户证书配置文件，并进行修改以满足您的要求：

```
# vi smime.cfg
```

有关可以在证书配置文件中配置哪些选项的详情，请查看 证书配置文件配置参数。

4. 更新现有证书配置文件：

```
# ipa certprofile-mod _profile_ID_ --file=smime.cfg
```

验证步骤

- 验证证书配置文件是否已更新：

```
$ ipa certprofile-show smime
  Profile ID: smime
  Profile description: New certificate profile description
  Store issued certificates: TRUE
```

其它资源

- 请参阅 ipa(a) 手册页。

- 请参阅 ipa help certprofile-mod。

### 64.7. 证书配置文件配置参数

证书配置文件配置参数存储在 CA 配置文件目录 /var/lib/pki/pki-tomcat/ca/profiles/ca 中的 *profile_name*.cfg 文件中。配置文件的所有参数 - 默认值、输入、输出和约束 - 都在单个策略集中配置。为证书配置集设置的策略具有名称 policyset.*policyName.policyNumber*。例如，对于策略设置 serverCertSet ：

```
policyset.list=serverCertSet
policyset.serverCertSet.list=1,2,3,4,5,6,7,8
policyset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl
policyset.serverCertSet.1.constraint.name=Subject Name Constraint
policyset.serverCertSet.1.constraint.params.pattern=CN=[^,]+,.+
policyset.serverCertSet.1.constraint.params.accept=true
policyset.serverCertSet.1.default.class_id=subjectNameDefaultImpl
policyset.serverCertSet.1.default.name=Subject Name Default
policyset.serverCertSet.1.default.params.name=CN=$request.req_subject_name.cn$, OU=pki-ipa, O=IPA
policyset.serverCertSet.2.constraint.class_id=validityConstraintImpl
policyset.serverCertSet.2.constraint.name=Validity Constraint
policyset.serverCertSet.2.constraint.params.range=740
policyset.serverCertSet.2.constraint.params.notBeforeCheck=false
policyset.serverCertSet.2.constraint.params.notAfterCheck=false
policyset.serverCertSet.2.default.class_id=validityDefaultImpl
policyset.serverCertSet.2.default.name=Validity Default
policyset.serverCertSet.2.default.params.range=731
policyset.serverCertSet.2.default.params.startTime=0
```

每个策略集都包含按照策略 ID 号为证书配置文件配置的策略列表，以它们的评估顺序排列。服务器为其收到的每个请求评估每个策略集。收到单个证书请求时，将评估一个集合，并忽略配置文件中的任何其

他集合。发布双密钥对后，对第一个证书请求评估第一个策略集，对第二个证书请求评估第二个策略集。在发布双密钥对时，在发布单个证书或多个集合时，您不需要多个策略集。

**表 64.1. 证书配置文件参数**

| 参数 | 描述 |
| --- | --- |
| desc | 证书配置文件的自由文本描述，显示在终端实体页面上。例如，**desc=This certificate profile 用于使用代理身份验证注册服务器证书。** |
| enable | 启用配置文件，使它可通过终端实体页面访问。例如：**enable=true**。 |
| auth.instance_id | 设置身份验证管理者插件，用来验证证书请求。要进行自动注册，如果身份验证成功，CA 会立即发布证书。如果身份验证失败或者没有指定身份验证插件，则会将请求排队，来由代理手动批准。例如，**auth.instance_id=AgentCertAuth**。 |
| authz.acl | 指定授权约束。这主要用于设置组评估访问控制列表 (ACL)。例如，**caCMCUserCert** 参数要求 CMC 请求的签名者属于证书管理者代理组：<br><br>**authz.acl=group="Certificate Manager Agents**<br><br>在基于目录的用户证书续订中，此选项用于确保原始请求者和当前验证的用户是同一个。在评估授权前，实体必须验证（绑定或登录到系统）。 |
| name | 证书配置文件的名称。例如，**name=Agent-Authenticated Server Certificate Enrollment**。此名称显示在最终用户注册或续订页面上。 |
| input.list | 按名称列出证书配置文件允许的输入。例如，**input.list=i1,i2**。 |
| input.input_id.class_id | 按输入 ID（在 input.list 中列出的输入名称）表示输入的 java 类名称。例如，**input.i1.class_id=certReqInputImpl**。 |
| output.list | 按名称列出证书配置文件可能的输出格式。例如 **output.list=o1**。 |
| output.output_id.class_id | 为在 output.list 中命名的输出格式指定 java 类名称。例如： **output.o1.class_id=certOutputImpl**。 |

| 参数 | 描述 |
| --- | --- |
| policyset.list | 列出配置的证书配置文件规则。对于双证书，一组规则适用于签名密钥，另一组规则适用于加密密钥。单个证书仅使用一组证书配置文件规则。例如，**policyset.list=serverCertSet**。 |
| policyset.policyset_id.list | 按照策略 ID 号，按评估的顺序，列出为证书配置文件配置的策略集合中的策略。例如：**policyset.serverCertSet.list=1,2,3,4,5,6,7,8**。 |
| policyset.policyset_id.policy_number.constraint.class_id | 表示配置文件规则中配置的默认约束插件集的 java 类名称。例如, policyset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl。 |
| policyset.policyset_id.policy_number.constraint.name | 提供用户定义的约束名称。例如，policyset.serverCertSet.1.constraint.name=Subject Name Constraint。 |
| policyset.policyset_id.policy_number.constraint.params.attribute | 为约束的允许的属性指定值。可能的属性因约束类型而异。例如, policyset.serverCertSet.1.constraint.params.pattern=CN=.*. |
| policyset.policyset_id.policy_number.default.class_id | 给出配置文件规则中默认集的 java 类名称。例如, policyset.serverCertSet.1.default.class_id=userSubjectNameDefaultImpl |
| policyset.policyset_id.policy_number.default.name | 给出用户定义的默认值的名称。例如：policyset.serverCertSet.1.default.name=Subject Name Default |
| policyset.policyset_id.policy_number.default.params.attribute | 为默认值的允许的属性指定值。可能的属性因默认类型而异。例如：policyset.serverCertSet.1.default.params.name=CN=(Name)$request.requestor_name$。 |

# 第 65 章 管理 IDM 中证书的有效性

在身份管理(IdM)中，您可以管理现有证书和未来要发布的证书的有效性，但方法有所不同。

## 65.1. 管理 IDM CA 发布的现有证书的有效性

在 IdM 中，可以使用以下方法查看证书的到期日期：

- 在 IdM WebUI 中查看到期日。
- 在 CLI 中查看到期日。

您可以使用以下方法管理 IdM CA 发布的现有证书的有效性：

- 通过使用原始证书签名请求(CSR)或私钥生成的新 CSR 请求新的证书来续订证书。您可以使用以下工具请求新证书：

  certmonger

  : 您可以使用 certmonger 请求服务证书。证书到期之前，certmonger 将自动续订证书，从而确保服务证书持续有效。详情请参阅 使用 certmonger 为服务获取 IdM 证书 。

  certutil

  : 您可以使用 certutil 续订用户、主机和服务证书。有关请求用户证书的详情，请参阅 请求新的用户证书并将其导出到客户端 ；

  openssl

  : 您可以使用 openssl 续订用户、主机和服务证书。

- 吊销证书。详情请查看：
  - 使用 IdM Web UI 吊销带有集成 IdM CA 的证书 ；

- 使用 IdM CLI 吊销带有集成 IdM CA 的证书 ；

- 如果证书已被临时吊销，则恢复证书。详情请查看：
  - 使用 IdM WebUI 恢复带有集成 IdM CA 的证书 ；
  - 使用 IdM CLI 恢复带有集成 IdM CA 的证书。

## 65.2. 管理 IDM CA 发布的未来证书的有效性

要管理 IdM CA 发布的未来证书的有效性，请修改、导入或创建证书配置文件。详情请参阅在 在身份管理中创建和管理证书配置文件。

## 65.3. 在 IDM WEBUI 中查看证书的到期日期

您可以使用 IdM WebUI 来查看 IdM CA 发布的所有证书的到期日期。

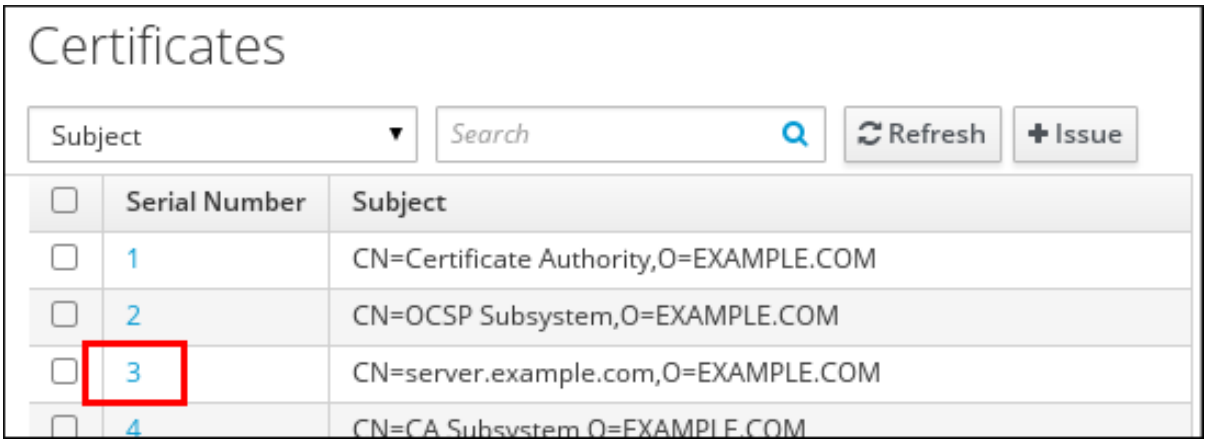
### 先决条件

- 确保您已获取管理员的凭证。

### 流程

1. 在 Authentication 菜单中，点击 Certificates > Certificates。
2. 单击证书的序列号，来打开证书信息页面。

图 65.1. 证书列表




3. 在证书信息页面中，找到 Expires On 信息。

## 65.4. 在 CLI 中查看证书的到期日期

您可以使用命令行界面(CLI)查看证书的到期日期。

**流程**

- 使用 openssl 工具以人类可读的格式打开文件：

```
$ openssl x509 -noout -text -in ca.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O = IDM.EXAMPLE.COM, CN = Certificate Authority
        Validity
            Not Before: Oct 30 19:39:14 2017 GMT
            Not After : Oct 30 19:39:14 2037 GMT
```

## 65.5. 吊销带有集成 IDM CA 的证书

### 65.5.1. 证书吊销原因

已吊销的证书是无效的，不能用于身份验证。所有取消都是永久的，除了原因 6：证书冻结。

默认的吊销原因为 0：未指定。

**表 65.1. 吊销原因**

| ID | 原因 | 解释 |
| --- | --- | --- |
| 0 | 未指定 | |
| 1 | 密钥泄露 | 签发证书的密钥不再被信任。<br>可能的原因是：丢失令牌，非正常访问文件。 |
| 2 | CA 泄露 | 签发证书的 CA 不再被信任。 |
| 3 | 隶属关系更改了 | 可能的原因：<br>* 本人已离开公司或转到另一个部门。<br>* 主机或服务将被停用。 |
| 4 | 被取代 | 较新的证书替换了当前的证书。 |
| 5 | 停止操作 | 主机或服务将被停用。 |
| 6 | 证书冻结 | 证书被临时吊销。您可稍后恢复证书。 |
| 8 | 从 CRL 中删除 | 证书不再包含在证书吊销列表(CRL)中。 |
| 9 | 特权收回 | 用户、主机或服务不再被允许使用证书。 |
| 10 | 属性授权(AA)泄露 | AA 证书不再被信任。 |

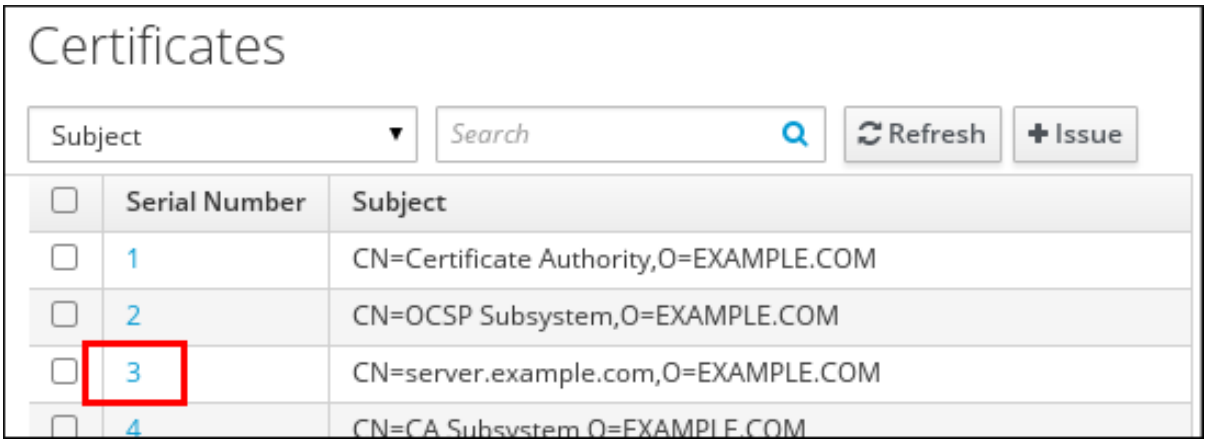
### 65.5.2. 使用 IdM Web UI 吊销带有集成 IdM CA 的证书

**如果您知道您已丢失证书的私钥，则您必须吊销证书以防止其被滥用。完成此流程，以使用 IdM WebUI 吊销 IdM CA 发布的证书。**

**流程**

1. **点击 Authentication > Certificates > Certificates。**
2. **单击证书的序列号，来打开证书信息页面。**

图 65.2. 证书列表




3. 在证书信息页面中，单击 **Actions** → **Revoke Certificate**。

4. 选择吊销的原因，然后单击 **Revoke**。详情请参阅 证书吊销原因。

### 65.5.3. 使用 IdM CLI 吊销带有集成 IdM CA 的证书

如果您知道您已丢失证书的私钥，则您必须吊销证书以防止其被滥用。完成此流程，以使用 IdM CLI 吊销 IdM CA 发布的证书。

**流程**

- 使用 ipa cert-revoke 命令，并指定：
  - 证书序列号
  - 吊销原因的 ID 号；有关详细信息，请参阅 证书吊销原因

例如，因为原因 1：密钥泄露，要吊销序列号为 1032 的证书，请输入：

```
$ ipa cert-revoke 1032 --revocation-reason=1
```

有关请求新证书的详情，请查看以下文档：

- 请求新的用户证书并将其导出到客户端

- 使用 certmonger 为服务获取 IdM 证书。

### 65.6. 恢复带有集成 IDM CA 的证书

如果您因为原因 6 ：证书冻结 吊销了证书 ， 如果证书的私钥未泄露， 您可以恢复它。要恢复证书， 请使用以下流程之一 ：

- 使用 IdM WebUI 恢复带有集成 IdM CA 的证书 ；

- 使用 IdM CLI 恢复带有集成 IdM CA 的证书。

#### 65.6.1. 使用 IdM WebUI 恢复带有集成 IdM CA 的证书

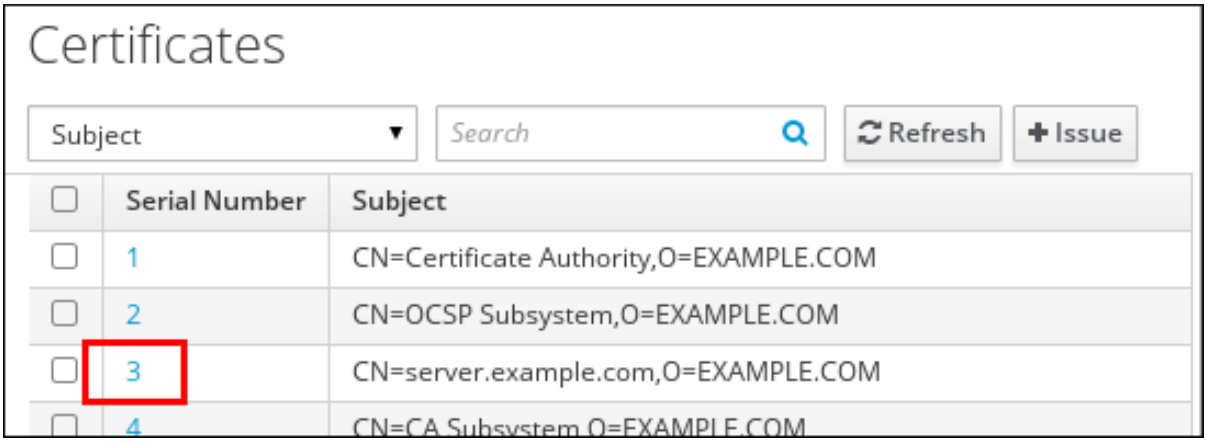
完成这个流程， 来使用 IdM WebUI 恢复因为原因 6 ：凭证冻结 而吊销的 IdM 证书。

**流程**

1. 在 Authentication 菜单中， 点击 Certificates > Certificates。

2. 单击证书的序列号， 来打开证书信息页面。

图 65.3. 证书列表



3. 在证书信息页面中，单击 **Actions → Restore Certificate**。

### 65.6.2. 使用 IdM CLI 恢复带有集成 IdM CA 的证书

**完成此流程，以使用 IdM CLI 恢复因为原因 6 ：证书冻结 而撤销的 IdM 证书。**

**流程**

- 使用 **ipa cert-remove-hold** 命令并指定证书序列号。例如：

```
$ ipa cert-remove-hold 1032
```

# 第 66 章 为智能卡验证配置身份管理

身份管理(IdM)支持使用如下方式的智能卡身份验证：

- IdM 证书颁发机构发布的用户证书
- 外部证书颁发机构发布的用户证书

您可以在 IdM 中为两种类型的证书配置智能卡验证。在这种情况下，rootca.pem CA 证书是包含可信外部证书颁发机构证书的文件。

有关 IdM 中智能卡验证的详情，请参考 了解智能卡验证。

有关配置智能卡验证的详情：

- 为智能卡验证配置 IdM 服务器
- 为智能卡验证配置 IdM 客户端
- 在 IdM Web UI 的用户条目中添加证书
- 在 IdM CLI 中向用户条目中添加证书
- 安装用来管理和使用智能卡的工具
- 在智能卡中存储证书
- 使用智能卡登录到 IdM

- 使用智能卡身份验证配置 GDM 访问
- 使用智能卡验证配置 su 访问

## 66.1. 为智能卡验证配置 IDM 服务器

如果要为配置了 IdM 服务器的 <EXAMPLE.ORG> 域的 <EXAMPLE.ORG> 域发布证书的用户启用智能卡验证，您必须获取以下证书，以便在运行配置 IdM 服务器的 ipa-advise 脚本时添加它们：

- 为 <EXAMPLE.ORG> CA 发布证书的根 CA 证书，或者通过一个或多个子CA 签发证书。您可以从认证机构发布的证书的网页下载证书链。详情请查看 配置浏览器来启用证书身份验证 中的步骤 1 - 4a。
- IdM CA 证书。您可以从运行 IdM CA 实例的 IdM 服务器上的 /etc/ipa/ca.crt 文件获取 CA 证书。
- 所有中间 CA 的证书，即 <EXAMPLE.ORG> CA 和 IdM CA 之间的中间。

为智能卡验证配置 IdM 服务器：

1. 以 PEM 格式获取 CA 证书的文件。
2. 运行内置的 ipa-advise 脚本。
3. 重新加载系统配置。

**先决条件**

- 有到 IdM 服务器的 root 访问权限。
- 您有 root CA 证书和所有中间 CA 证书。

**流程**

1. 创建要进行配置的目录：

```
[root@server]# mkdir ~/SmartCard/
```

2. 进入该目录：

```
[root@server]# cd ~/SmartCard/
```

3. 获取存储在 PEM 格式文件中的相关 CA 证书。如果您的 CA 证书存储再不同格式的文件中，如 DER，请将其转换为 PEM 格式。IdM 证书颁发机构证书采用 PEM 格式，位于 /etc/ipa/ca.crt 文件中。

   将 DER 文件转换为 PEM 文件：

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

4. 为方便起见，将证书复制到您要进行配置的目录中：

```
[root@server SmartCard]# cp /tmp/rootca.pem ~/SmartCard/
[root@server SmartCard]# cp /tmp/subca.pem ~/SmartCard/
[root@server SmartCard]# cp /tmp/issuingca.pem ~/SmartCard/
```

5. 另外，如果您使用外部证书颁发机构的证书，请使用 openssl x509 工具查看 PEM 格式的文件内容，来检查 Issuer 和 Subject 值是否正确：

```
[root@server SmartCard]# openssl x509 -noout -text -in rootca.pem | more
```

6. 使用管理员特权，通过内置的 ipa-advise 工具生成配置脚本：

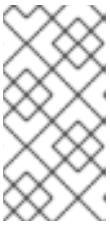
```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# ipa-advise config-server-for-smart-card-auth > config-
server-for-smart-card-auth.sh
```

   config-server-for-smart-card-auth.sh 脚本执行以下操作：

- 它配置 IdM Apache HTTP 服务器。
- 它在 KDC（Key Distribution Center）中启用 PKINIT（Public Key Cryptography for Initial Authentication in Kerberos）。
- 它将 IdM Web UI 配置为接受智能卡授权请求。

7. 执行脚本，将包含根 CA 和子 CA 证书的 PEM 文件添加为参数：

```
[root@server SmartCard]# chmod +x config-server-for-smart-card-auth.sh
[root@server SmartCard]# ./config-server-for-smart-card-auth.sh rootca.pem
subca.pem issuingca.pem
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



**注意**

在任何子 CA 证书前，确保将根 CA 的证书添加为参数，并且 CA 或子 CA 证书还没有过期。

8. 另外，如果发布用户证书的证书颁发机构不提供任何在线证书状态协议(OCSP)响应程序，您可能需要禁用对 IdM Web UI 身份验证的 OCSP 检查：

a. 在 /etc/httpd/conf.d/ssl.conf 文件中将 SSLOCSPEnable 参数设为 off ：

```
SSLOCSPEnable off
```

b. 重启 Apache 守护进程(httpd)使更改立即生效：

```
[root@server SmartCard]# systemctl restart httpd
```

**警告**

如果您只使用 IdM CA 发出的用户证书，不要禁用 OCSP 检查。OCSP 响应器是 IdM 的一部分。

有关如何保持 OCSP 检查处于启用状态，同时防止 IdM 服务器拒绝用户证书（如果 IdM 服务器不包含有关颁发用户证书的 CA 侦听 OCSP 服务请求的位置的信息）的说明，请参阅 Apache mod_ssl 配置选项 中的 SSLOCSPDefaultResponder 指令。

该服务器现在被配置为智能卡验证。

**注意**

要在整个拓扑中启用智能卡验证，请在每个 IdM 服务器中运行操作过程。

## 66.2. 使用 ANSIBLE 为智能卡验证配置 IDM 服务器

您可以使用 Ansible 为已由 Identity Management (IdM) CA 信任的 <EXAMPLE.ORG> 域发布的 CA 的用户启用智能卡验证。要做到这一点，您必须获取以下证书，以便在使用 ipasmartcard_server ansible-freeipa 角色脚本运行 Ansible playbook 时使用它们：

- 为 <EXAMPLE.ORG> CA 发布证书的根 CA 证书，或者通过一个或多个子CA 签发证书。您可以从认证机构发布的证书的网页下载证书链。详情请参阅 配置浏览器中的步骤 4 以启用证书验证。
- IdM CA 证书。您可以从任何 IdM CA 服务器上的 /etc/ipa/ca.crt 文件获取 CA 证书。
- <EXAMPLE.ORG> CA 和 IdM CA 之间的中间所有 CA 的证书。

**先决条件**

- 您有访问 IdM 服务器的 root 权限。

- 您需要知道 IdM admin 密码。
- 您有 root CA 证书、IdM CA 证书以及所有中间 CA 证书。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 如果您的 CA 证书以不同格式（如 DER ）的文件存储，请将其转换为 PEM 格式：

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

IdM 证书颁发机构证书采用 PEM 格式，并位于 /etc/ipa/ca.crt 文件中。

2. （可选）使用 openssl x509 工具查看 PEM 格式文件的内容，以检查 Issuer 和 Subject的值是否正确：

```
# openssl x509 -noout -text -in root-ca.pem | more
```

3. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

4. 创建专用于 CA 证书的子目录：

```
$ mkdir SmartCard/
```

5. 为方便起见，将所有必需的证书复制到 ~/MyPlaybooks/SmartCard/ 目录中：

```
# cp /tmp/root-ca.pem ~/MyPlaybooks/SmartCard/
# cp /tmp/intermediate-ca.pem ~/MyPlaybooks/SmartCard/
# cp /etc/ipa/ca.crt ~/MyPlaybooks/SmartCard/ipa-ca.crt
```

6. 在 Ansible 清单文件中指定以下内容：

- 要为智能卡验证配置的 IdM 服务器。
- IdM 管理员密码。
- CA 证书的路径按以下顺序：
  - root CA 证书文件
  - 中间 CA 证书文件
  - IdM CA 证书文件

文件可以类似如下：

```
[ipaserver]
ipaserver.idm.example.com

[ipareplicas]
ipareplica1.idm.example.com
ipareplica2.idm.example.com
```

```
[ipacluster:children]
ipaserver
ipareplicas

[ipacluster:vars]
ipaadmin_password= "{{ ipaadmin_password }}"
ipasmartcard_server_ca_certs=/home/<user_name>/MyPlaybooks/SmartCard/root-
ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/intermediate-
ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/ipa-ca.crt
```

7. 使用以下内容创建 install-smartcard-server.yml playbook：

```
---
- name: Playbook to set up smart card authentication for an IdM server
  hosts: ipaserver
  become: true

  roles:
  - role: ipasmartcard_server
    state: present
```

8. 保存该文件。

9. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory install-
smartcard-server.yml
```

ipasmartcard_server Ansible 角色执行以下操作：

- 它配置 IdM Apache HTTP 服务器。
- 它在 KDC（Key Distribution Center）中启用 PKINIT（Public Key Cryptography for Initial Authentication in Kerberos）。
- 它将 IdM Web UI 配置为接受智能卡授权请求。

10. 另外，如果发布用户证书的证书颁发机构不提供任何在线证书状态协议(OCSP)响应程序，您可能需要禁用对 IdM Web UI 身份验证的 OCSP 检查：

a. 以 root 用户身份连接到 IdM 服务器：

```
ssh root@ipaserver.idm.example.com
```

b. 在 /etc/httpd/conf.d/ssl.conf 文件中将 SSLOCSPEnable 参数设为 off ：

```
SSLOCSPEnable off
```

c. 重启 Apache 守护进程(httpd)使更改立即生效：

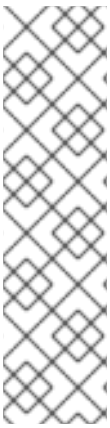
```
# systemctl restart httpd
```

**警告**

如果您只使用 IdM CA 发出的用户证书，不要禁用 OCSP 检查。OCSP 响应器是 IdM 的一部分。

有关如何保持 OCSP 检查处于启用状态，同时防止 IdM 服务器拒绝用户证书（如果 IdM 服务器不包含有关颁发用户证书的 CA 侦听 OCSP 服务请求的位置的信息）的说明，请参阅 Apache mod_ssl 配置选项 中的 SSLOCSPDefaultResponder 指令。

清单文件中列出的服务器现在被配置为智能卡验证。



**注意**

要在整个拓扑中启用智能卡验证，将 Ansible playbook 中的 hosts 变量设置为 ipacluster ：

```
---
- name: Playbook to setup smartcard for IPA server and replicas
  hosts: ipacluster
[...]
```

**其它资源**

- 在 /usr/share/doc/ansible-freeipa/playbooks/ 目录中使用 ipasmartcard_server 角色的 playbook 示例

## 66.3. 为智能卡验证配置 IDM 客户端

按照以下流程为智能卡验证配置 IdM 客户端。这个过程需要运行在每个 IdM 系统、客户端或服务器上，您希望在使用智能卡进行身份验证时连接到这些系统。例如，若要启用从主机 A 到主机 B 的 ssh 连接，需要在主机 B 上运行脚本。

作为管理员，运行这个流程来使用如下方法启用智能卡身份验证

- ssh 协议

  详情请查看 使用智能卡验证配置 SSH 访问。

- 控制台登录
- GNOME 显示管理器(GDM)
- su 命令

对于向 IdM Web UI 进行身份验证，不需要此流程。向 IdM Web UI 进行身份验证涉及两个主机，它们都不必是 IdM 客户端：

- 运行浏览器的机器。机器可以在 IdM 域之外。
- 在其上运行 httpd 的 IdM 服务器。

以下流程假设您在 IdM 客户端，而不是 IdM 服务器上配置智能卡身份验证。因此，您需要两台计算机：生成配置脚本的 IdM 服务器，以及运行脚本的 IdM 客户端。

先决条件

- 已经为智能卡验证配置了 IdM 服务器，如 为智能卡验证配置 IdM 服务器 中所述。
- 有对 IdM 服务器和 IdM 客户端的 root 访问权限。
- 您有 root CA 证书和所有中间 CA 证书。
- 您使用 --mkhomedir 选项安装了 IdM 客户端，以确保远程用户可以成功登录。如果您没有创建主目录，则默认登录位置为目录结构的根目录 /。

**流程**

1. 在 IdM 服务器上，使用管理员权限通过 ipa-advise 生成配置脚本：

```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# ipa-advise config-client-for-smart-card-auth > config-client-for-smart-card-auth.sh
```

config-client-for-smart-card-auth.sh 脚本执行以下操作：

- 它配置智能卡守护进程。
- 它设置系统范围的信任存储。
- 它配置系统安全服务守护进程(SSSD)，允许用户使用其用户名和密码或其智能卡进行验证。有关智能卡验证的 SSSD 配置集选项的详情，请参考 RHEL 中的智能卡验证选项。

2. 从 IdM 服务器中，将脚本复制到 IdM 客户端机器中选择的目录中：

```
[root@server SmartCard]# scp config-client-for-smart-card-auth.sh root@client.idm.example.com:/root/SmartCard/
Password:
config-client-for-smart-card-auth.sh        100%  2419     3.5MB/s   00:00
```

3. 从 IdM 服务器中，使用 PEM 格式将 CA 证书文件复制到上一步中使用的 IdM 客户端机器中的同一目录中：

```
[root@server SmartCard]# scp {rootca.pem,subca.pem,issuingca.pem}
root@client.idm.example.com:/root/SmartCard/
Password:
rootca.pem                    100%  1237     9.6KB/s   00:00
subca.pem                     100%  2514    19.6KB/s   00:00
issuingca.pem                  100%  2514    19.6KB/s   00:00
```

4. 在客户端机器上执行脚本，将包含 CA 证书的 PEM 文件添加为参数：

```
[root@client SmartCard]# kinit admin
[root@client SmartCard]# chmod +x config-client-for-smart-card-auth.sh
[root@client SmartCard]# ./config-client-for-smart-card-auth.sh rootca.pem subca.pem
issuingca.pem
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```

**注意**

在任何子 CA 证书前，确保将根 CA 的证书添加为参数，并且 CA 或子 CA 证书还没有过期。

现在为智能卡验证配置了客户端。

## 66.4. 使用 ANSIBLE 为智能卡验证配置 IDM 客户端

按照以下流程，使用 ansible-freeipa ipasmartcard_client 模块配置特定的身份管理(IdM)客户端，以允许 IdM 用户使用智能卡进行身份验证。运行这个步骤为使用以下任一方法的 IdM 用户启用智能卡验证：

- ssh 协议

  详情请查看 使用智能卡验证配置 SSH 访问。

- 控制台登录

- GNOME 显示管理器(GDM)
- su 命令

**注意**

对于向 IdM Web UI 进行身份验证，不需要此流程。向 IdM Web UI 进行身份验证涉及两个主机，它们都不必是 IdM 客户端：

- 运行浏览器的机器。机器可以在 IdM 域之外。
- 在其上运行 httpd 的 IdM 服务器。

**先决条件**

- 为智能卡验证配置了您的 IdM 服务器，如使用 Ansible 配置 IdM 服务器进行智能卡验证 所述。
- 有对 IdM 服务器和 IdM 客户端的 root 访问权限。
- 您有 root CA 证书、IdM CA 证书以及所有中间 CA 证书。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 如果您的 CA 证书以不同格式（如 DER ）的文件存储，请将其转换为 PEM 格式：

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

IdM CA 证书采用 PEM 格式，并位于 /etc/ipa/ca.crt 文件中。

2. （可选）使用 openssl x509 工具查看 PEM 格式文件的内容，以检查 Issuer 和 Subject的值是否正确：

```
# openssl x509 -noout -text -in root-ca.pem | more
```

3. 在 Ansible 控制节点上，导航到 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

4. 创建专用于 CA 证书的子目录：

```
$ mkdir SmartCard/
```

5. 为方便起见，将所有必需的证书复制到 ~/MyPlaybooks/SmartCard/ 目录中，例如：

```
# cp /tmp/root-ca.pem ~/MyPlaybooks/SmartCard/
# cp /tmp/intermediate-ca.pem ~/MyPlaybooks/SmartCard/
# cp /etc/ipa/ca.crt ~/MyPlaybooks/SmartCard/ipa-ca.crt
```

6. 在 Ansible 清单文件中指定以下内容：

- 要为智能卡验证配置的 IdM 客户端。
- IdM 管理员密码。
- CA 证书的路径按以下顺序：
  - root CA 证书文件
  - 中间 CA 证书文件
  - IdM CA 证书文件

文件可以类似如下：

```
[ipaclients]
ipaclient1.example.com
ipaclient2.example.com

[ipaclients:vars]
ipaadmin_password=SomeADMINpassword
ipasmartcard_client_ca_certs=/home/<user_name>/MyPlaybooks/SmartCard/root-
ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/intermediate-
ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/ipa-ca.crt
```

7. 使用以下内容创建 install-smartcard-clients.yml playbook：

```
---
- name: Playbook to set up smart card authentication for an IdM client
  hosts: ipaclients
  become: true

  roles:
  - role: ipasmartcard_client
    state: present
```

8. 保存该文件。

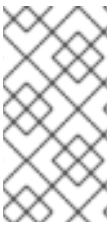
9. 运行 Ansible playbook。指定 playbook 和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory install-smartcard-clients.yml
```

ipasmartcard_client Ansible 角色执行以下操作：

- 它配置智能卡守护进程。
- 它设置系统范围的信任存储。
- 它将系统安全服务守护进程(SSSD)配置为允许用户通过其用户名和密码或者智能卡进行身份验证。有关智能卡验证的 SSSD 配置集选项的详情，请参考 RHEL 中的智能卡验证选项。

现在为智能卡验证配置了清单文件的 ipaclients 部分中列出的客户端。



注意

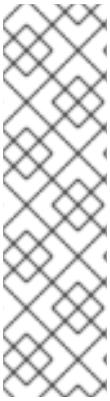
如果您使用 --mkhomedir 选项安装了 IdM 客户端，远程用户将能够登录到其主目录。否则，默认登录位置是目录结构 / 的根目录。

其它资源

- 在 /usr/share/doc/ansible-freeipa/playbooks/ 目录中使用 ipasmartcard_server 角色的 playbook 示例

## 66.5. 在 IDM WEB UI 的用户条目中添加证书

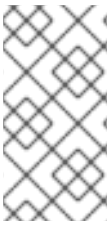
按照以下流程，向 IdM Web UI 中的用户条目中添加外部证书。



**注意**

也可以将证书映射数据上传到 IdM 中的用户条目，而不必上传整个证书。包含完整证书或证书映射数据的用户条目可以和相应的证书映射规则一起使用，以便于系统管理员配置智能卡身份验证。详情请查看

配置身份验证的证书映射规则。



**注意**

如果用户的证书由 IdM 证书颁发机构发布，则证书已存储在用户条目中，您不需要按照此流程操作。

**先决条件**

- 您有要添加到用户条目的证书。

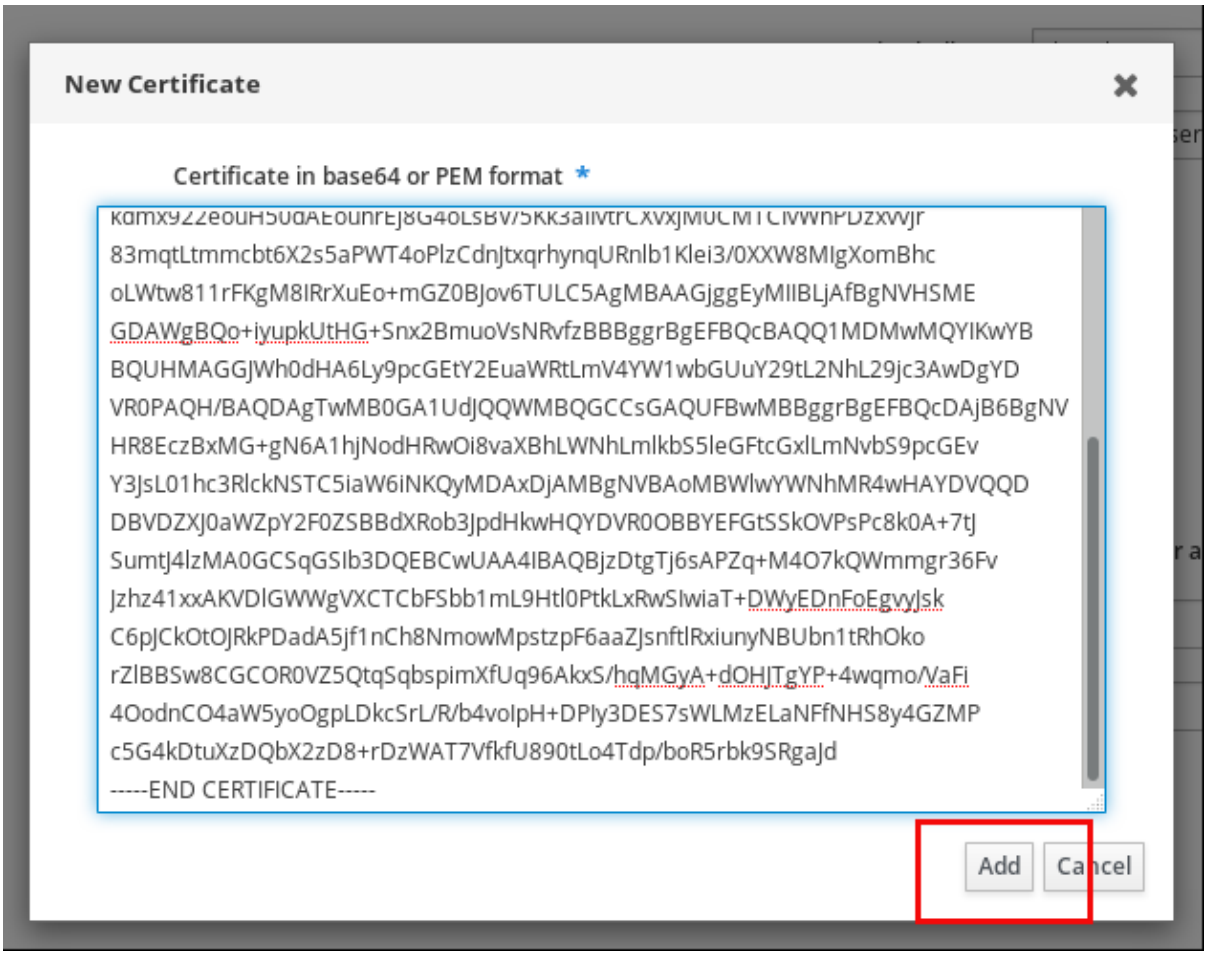
**流程**

1. 如果要给另一个用户添加证书，请以管理员身份登录到 IdM Web UI。要在您自己的配置文件中添加证书，您不需要管理员的凭证。
2. 导航到 **Users** → **Active users** → **sc_user**。
3. 找到 **Certificate** 选项，并单击 **Add**。
4. 在命令行界面中，使用 **cat** 工具或文本编辑器以 PEM 格式显示证书：

```
[user@client SmartCard]$ cat testuser.crt
```

5. 将证书从 CLI 复制并粘贴到 Web UI 中打开的窗口中。
6. 点 **Add**。

图 66.1. 在 IdM Web UI 中添加新证书




**sc_user 条目现在包含一个外部证书。**

## 66.6. 在 IDM CLI 中向用户条目中添加证书

**按照以下流程，将外部证书添加到 IdM CLI 中的用户条目。**

**注意**

**也可以将证书映射数据上传到 IdM 中的用户条目，而不必上传整个证书。包含完整证书或证书映射数据的用户条目可以和相应的证书映射规则一起使用，以便于系统管理员配置智能卡身份验证。详情请参阅 配置身份验证的证书映射规则。**

**注意**

**如果用户的证书由 IdM 证书颁发机构发布，则证书已存储在用户条目中，您不需要按照此流程操作。**

**先决条件**

- 您有要添加到用户条目的证书。

**流程**

1. 如果要给另一个用户添加证书，请以管理员身份登录到 IdM CLI：

```
[user@client SmartCard]$ kinit admin
```

要在您自己的配置文件中添加证书，您不需要管理员的凭证：

```
[user@client SmartCard]$ kinit sc_user
```

2. 创建一个包含证书的环境变量，该变量移除了标头和页脚，并串联成一行，这是 ipa user-add-cert 命令期望的格式：

```
[user@client SmartCard]$ export CERT=`openssl x509 -outform der -in testuser.crt | base64 -w0 -`
```

请注意，testuser.crt 文件中的证书必须是 PEM 格式。

3. 使用 ipa user-add-cert 命令将证书添加到 sc_user 的配置文件：

```
[user@client SmartCard]$ ipa user-add-cert sc_user --certificate=$CERT
```

sc_user 条目现在包含一个外部证书。

## 66.7. 安装用来管理和使用智能卡的工具

**先决条件**

- gnutls-utils 软件包已安装。
- opensc 软件包已安装。

- pcscd 服务正在运行。

在配置智能卡前，您必须安装相应的工具，该工具可以生成证书，并启动 pscd 服务。

流程

1. 安装 opensc 和 gnutls-utils 软件包：

```
# yum -y install opensc gnutls-utils
```

2. 启动 pcscd 服务。

```
# systemctl start pcscd
```

验证步骤

- 验证 pcscd 服务是否已启动并运行

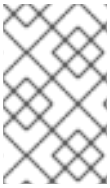
```
# systemctl status pcscd
```

## 66.8. 准备智能卡并将证书和密钥上传到智能卡

按照以下流程，使用 pkcs15-init 工具配置智能卡，该工具可帮助您配置：

- 擦除智能卡
- 设置新的 PIN 和可选的 PIN Unblocking Keys（PUKs）
- 在智能卡上创建新插槽
- 在插槽存储证书、私钥和公钥

- 如果需要，请锁定智能卡设置，因为某些智能卡需要这个类型的最终化



**注意**

pkcs15-init 工具可能无法使用所有智能卡。您必须使用您使用智能卡的工具。

**先决条件**

- 已安装 opensc 软件包，其中包括 pkcs15-init 工具。

  如需了解更多详细信息，请参阅 安装用于管理和使用智能卡的工具。

- 该卡插入读卡器并连接到计算机。

- 您有一个要存储在智能卡上的私钥、公钥和证书。在此流程中，testuser.key、testuserpublic.key 和 testuser.crt 是用于私钥、公钥和证书的名称。

- 您有当前的智能卡用户 PIN 和 Security Officer PIN(SO-PIN)。

**流程**

1. 擦除智能卡并使用您的 PIN 验证自己：

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

   这个卡已经被清除。

2. 初始化智能卡，设置您的用户 PIN 和 PUK，以及您的安全响应 PIN 和 PUK：

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \ --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

**pcks15-init 工具在智能卡上创建一个新插槽。**

3.

**为插槽设置标签和验证 ID：**

```
$ pkcs15-init --store-pin --label testuser \ --auth-id 01 --so-pin 65498714 --pin 963214 --puk
321478
Using reader with a card: Reader name
```

**标签设置为人类可读的值，在本例中为 testuser。auth-id 必须是两个十六进制值，在本例中设为 01。**

4.

**在智能卡的新插槽中存储并标记私钥：**

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \ --auth-id 01 --id 01 --pin
963214
Using reader with a card: Reader name
```

**注意**

**在存储您的私钥并将证书存储在下一步中时，您为 --id 指定的值必须相同。建议为 --id 指定自己的值，否则它们将更复杂的值由工具计算。**

5.

**在智能卡上的新插槽中存储并标记该证书：**

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \ --auth-id 01 --id 01 --format
pem --pin 963214
Using reader with a card: Reader name
```

6.

**可选：在智能卡的新插槽中保存并标记公钥：**

```
$ pkcs15-init --store-public-key testuserpublic.key --label testuserpublic_key --auth-id 01 --id
01 --pin 963214
Using reader with a card: Reader name
```

**注意**

**如果公钥与私钥或证书对应，请指定与私钥或证书的 ID 相同的 ID。**

7. 可选：某些智能卡要求您通过锁定设置来完成卡：

```
$ pkcs15-init -F
```

此时您的智能卡在新创建的插槽中包含证书、私钥和公钥。您还创建了您的用户 PIN 和 PUK，以及安全响应 PIN 和 PUK。

## 66.9. 使用智能卡登录到 IDM

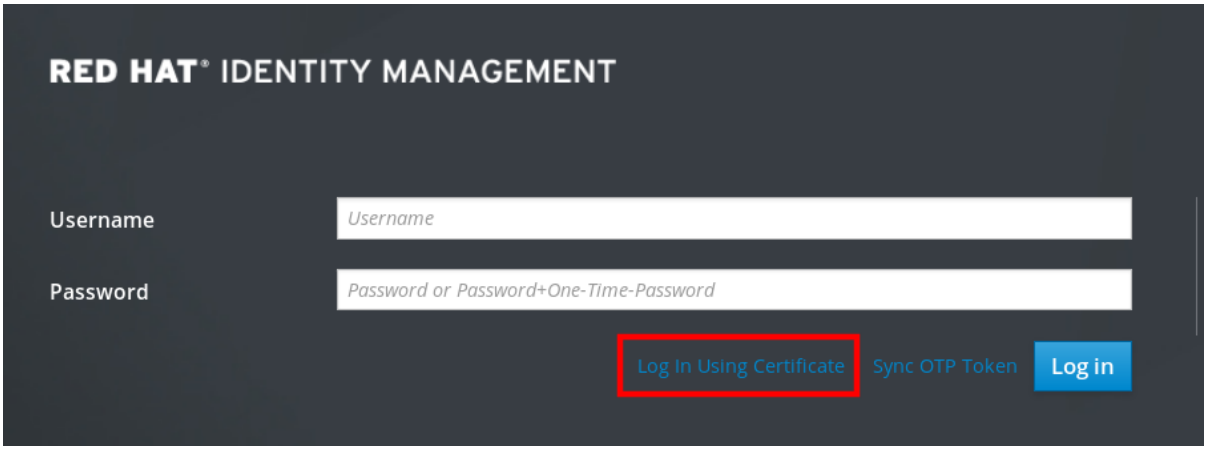
按照以下流程，使用智能卡登录到 IdM Web UI。

### 先决条件

- web 浏览器被配置为使用智能卡验证。
- IdM 服务器被配置为智能卡验证。
- 在您的智能卡中安装的证书由 IdM 服务器发出，或者已添加到 IdM 的用户条目中。
- 您知道解锁智能卡所需的 PIN。
- 智能卡已插入到读取器中。

### 流程

1. 在浏览器中打开 IdM Web UI。
2. 点 Log In 使用证书。



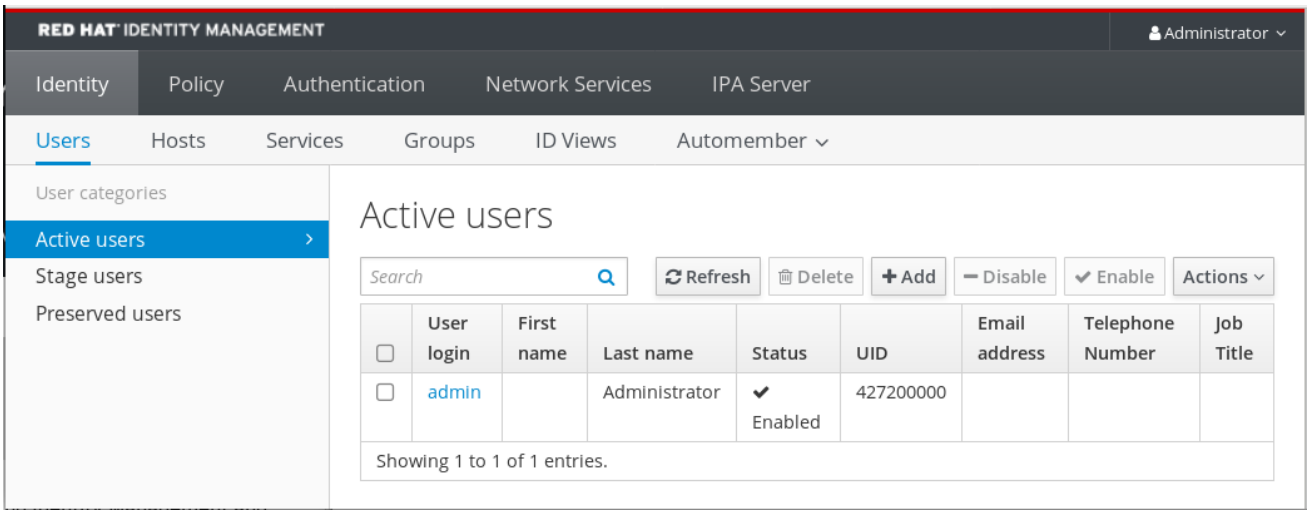

3. **如果 Password Required 对话框打开，请添加 PIN 来解锁智能卡，然后单击 OK 按钮。**

   **此时会打开 User Identification Request 对话框。**

   **如果智能卡包含多个证书，请在 选择用于验证的证书 下方的下拉列表中选择您要用于身份验证的证书。**

4. **单击 OK 按钮。**

**现在，您已成功登录到 IdM Web UI。**




## 66.10. 在 IDM 客户端中使用智能卡验证登录到 GDM

GNOME 桌面管理器(GDM)需要身份验证。您可以使用您的密码，但也可以使用智能卡进行验证。

按照以下流程，使用智能卡验证来访问 GDM。

先决条件

- 为智能卡验证配置了系统。详情请参阅 为智能卡验证配置 IdM 客户端。
- 该智能卡包含您的证书和私钥。
- 该用户帐户是 IdM 域的成员。
- 智能卡上的证书通过以下方式映射到用户条目：
  - 为特定用户条目分配证书。详情请参阅在 在 IdM Web UI 中将证书添加给用户条目 或 在 IdM CLI 中将证书添加给用户条目。
  - 应用到该帐户的证书映射数据。详情请查看 在智能卡上配置身份验证的证书映射规则。

流程

1. 在读取器中插入智能卡。
2. 输入智能卡 PIN。
3. 点 Sign In。

您成功登录到 RHEL 系统，并且您有一张由 IdM 服务器提供的 TGT。

验证步骤

- 在 Terminal 中输入 klist，并检查结果：

```
$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: example.user@REDHAT.COM

Valid starting       Expires              Service principal
04/20/2020 13:58:24  04/20/2020 23:58:24  krbtgt/EXAMPLE.COM@EXAMPLE.COM
 renew until 04/27/2020 08:58:15
```

## 66.11. 在 SU 命令中使用智能卡验证

切换到其他用户需要身份验证。您可以使用密码或证书。按照以下流程，通过 su 命令使用智能卡。这意味着输入 su 命令后，系统会提示您输入智能卡 PIN。

### 先决条件

- 为智能卡验证配置了您的 IdM 服务器和客户端。
  - 请参阅为智能卡验证配置 IdM 服务器
  - 请参阅为智能卡验证配置 IdM 客户端
- 该智能卡包含您的证书和私钥。请参阅智能卡中的证书
- 该卡插入读卡器并连接到计算机。

### 流程

- 在终端窗口中，使用 su 命令切换到其他用户：

```
$ su - example.user
PIN for smart_card
```

如果配置正确，会提示您输入智能卡 PIN。

# 第 67 章 为 IDM 中智能卡验证配置 ADCS 发布的证书

要在 IdM 中为其证书是由活动目录(AD)证书服务发布的用户配置智能卡验证：

- 您的部署是基于身份管理(IdM)和活动目录(AD)之间的跨林信任。
- 您希望允许智能卡验证存储在 AD 中的帐户的用户。
- 证书创建并存储在活动目录证书服务(ADCS)中。

有关智能卡验证的概述，请参阅了解智能卡验证。

配置通过以下步骤完成：

- 将 CA 和用户证书从活动目录复制到 IdM 服务器和客户端
- 使用 ADCS 证书为智能卡身份验证配置 IdM 服务器和客户端
- 转换 PFX(PKCS#12)文件，以便能够将证书和私钥存储到智能卡中
- 在 sssd.conf 文件中配置超时
- 为智能卡身份验证创建证书映射规则

先决条件

- 身份管理(IdM)和活动目录(AD)信任已安装

  详情请参阅在 IdM 和 AD 之间安装信任。

- 活动目录证书服务(ADCS)已安装，并且用户证书已生成

## 67.1. 信任配置和证书使用量所需的 WINDOWS 服务器设置

您必须在 Windows 服务器上配置以下内容：

- 已安装活动目录证书服务(ADCS)
- 创建证书颁发机构
- [可选] 如果您正在使用证书颁发机构 Web 注册，则必须配置互联网信息服务(IIS)

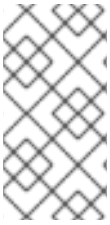
导出证书：

- 密钥必须有 2048 位或更多
- 包括一个私钥
- 您将需要以下格式的证书：个人信息交换— PKCS #12(.PFX)
  - 启用证书隐私

## 67.2. 使用 SFTP 从 ACTIVE DIRECTORY 复制证书

要能够使用智能卡身份验证，您需要复制以下证书文件：

- CER 格式的根 CA 证书：IdM 服务器上的 adcs-winserver-ca.cer。
- 具有 PFX 格式私钥的用户证书：IdM 客户端上的 aduser1.pfx。



**注意**

这个过程预期 SSH 访问是允许的。如果 SSH 不可用，用户必须将 文件从 AD 服务器复制到 IdM 服务器和客户端。

**流程**

1. 从 IdM 服务器 连接，并将 adcs-winserver-ca.cer 根证书复制到 IdM 服务器：

```
root@idmserver ~]# sftp Administrator@winserver.ad.example.com
Administrator@winserver.ad.example.com's password:
Connected to Administrator@winserver.ad.example.com.
sftp> cd <Path to certificates>
sftp> ls
adcs-winserver-ca.cer    aduser1.pfx
sftp>
sftp> get adcs-winserver-ca.cer
Fetching <Path to certificates>/adcs-winserver-ca.cer to adcs-winserver-ca.cer
<Path to certificates>/adcs-winserver-ca.cer                100%  1254    15KB/s 00:00
sftp quit
```

2. 从 IdM 客户端 连接，并将 aduser1.pfx 用户证书复制到客户端：

```
[root@client1 ~]# sftp Administrator@winserver.ad.example.com
Administrator@winserver.ad.example.com's password:
Connected to Administrator@winserver.ad.example.com.
sftp> cd /<Path to certificates>
sftp> get aduser1.pfx
Fetching <Path to certificates>/aduser1.pfx to aduser1.pfx
<Path to certificates>/aduser1.pfx                100%  1254    15KB/s 00:00
sftp quit
```

现在，CA 证书保存在 IdM 服务器上，用户证书存储在客户端机器上。

### 67.3. 使用 ADCS 证书为智能卡身份验证配置 IDM 服务器和客户端

您必须配置 IdM（身份管理）服务器和客户端，以便能够在 IdM 环境中使用智能卡身份验证。IdM 包含进行了所有必要更改的 ipa-advise 脚本：

- 安装所需的软件包

- 配置 IdM 服务器和客户端
- 将 CA 证书复制到预期的位置

您可以在 IdM 服务器中运行 ipa-advise。

按照以下流程为智能卡验证配置服务器和客户端：

- 在 IdM 服务器中：准备 ipa-advise 脚本，为智能卡验证配置 IdM 服务器。
- 在 IdM 服务器中：准备 ipa-advise 脚本，以配置 IdM 客户端以进行智能卡验证。
- 在 IdM 服务器中：使用 AD 证书应用 IdM 服务器上的 ipa-advise 服务器脚本。
- 将客户端脚本移动到 IdM 客户端机器中。
- 在 IdM 客户端上：使用 AD 证书在 IdM 客户端上应用 ipa-advise 客户端脚本。

**先决条件**

- 证书已复制到 IdM 服务器。
- 获取 Kerberos 票据。
- 以具有管理权限的用户身份登录。

**流程**

1. 在 IdM 服务器上，使用 ipa-advise 脚本来配置客户端：

```
[root@idmserver ~]# ipa-advise config-client-for-smart-card-auth > sc_client.sh
```

2. 在 IdM 服务器上，使用 ipa-advise 脚本来配置服务器：

```
[root@idmserver ~]# ipa-advise config-server-for-smart-card-auth > sc_server.sh
```

3. 在 IdM 服务器中执行脚本：

```
[root@idmserver ~]# sh -x sc_server.sh adcs-winserver-ca.cer
```

- 它配置 IdM Apache HTTP 服务器。
- 它在 KDC（Key Distribution Center）中启用 PKINIT（Public Key Cryptography for Initial Authentication in Kerberos）。
- 它将 IdM Web UI 配置为接受智能卡授权请求。

4. 将 sc_client.sh 脚本复制到客户端系统中：

```
[root@idmserver ~]# scp sc_client.sh root@client1.idm.example.com:/root
Password:
sc_client.sh                  100%  2857   1.6MB/s   00:00
```

5. 将 Windows 证书复制到客户端系统中：

```
[root@idmserver ~]# scp adcs-winserver-ca.cer root@client1.idm.example.com:/root
Password:
adcs-winserver-ca.cer                  100%  1254   952.0KB/s   00:00
```

6. 在客户端系统中运行客户端脚本：

```
[root@idmclient1 ~]# sh -x sc_client.sh adcs-winserver-ca.cer
```

CA 证书以正确格式安装在 IdM 服务器和客户端系统中，下一步是将用户证书复制到智能卡本身。

## 67.4. 转换 PFX 文件

在将 PFX(PKCS#12)文件保存到智能卡中前，您必须：

- 将文件转换为 PEM 格式
- 将私钥和证书提取到两个不同的文件

**先决条件**

- PFX 文件被复制到 IdM 客户端机器中。

**流程**

1. 在 IdM 客户端中，采用 PEM 格式：

```
[root@idmclient1 ~]# openssl pkcs12 -in aduser1.pfx -out aduser1_cert_only.pem -
clcerts -nodes
Enter Import Password:
```

2. 将密钥提取到单独的文件中：

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -nocerts -out adduser1.pem >
aduser1.key
```

3. 将公共证书提取到单独的文件中：

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -clcerts -nokeys -out
aduser1_cert_only.pem > aduser1.crt
```

此时，您可以将 aduser1.key 和 aduser1.crt 存储在智能卡中。

## 67.5. 安装用来管理和使用智能卡的工具

**先决条件**

- gnutls-utils 软件包已安装。
- opensc 软件包已安装。
- pcscd 服务正在运行。

在配置智能卡前，您必须安装相应的工具，该工具可以生成证书，并启动 pscd 服务。

流程

1. 安装 opensc 和 gnutls-utils 软件包：

```
# yum -y install opensc gnutls-utils
```

2. 启动 pcscd 服务。

```
# systemctl start pcscd
```

验证步骤

- 验证 pcscd 服务是否已启动并运行

```
# systemctl status pcscd
```

### 67.6. 准备智能卡并将证书和密钥上传到智能卡

按照以下流程，使用 pkcs15-init 工具配置智能卡，该工具可帮助您配置：

- 擦除智能卡
- 设置新的 PIN 和可选的 PIN Unblocking Keys（PUKs）

- 在智能卡上创建新插槽
- 在插槽存储证书、私钥和公钥
- 如果需要，请锁定智能卡设置，因为某些智能卡需要这个类型的最终化

**注意**

pkcs15-init 工具可能无法使用所有智能卡。您必须使用您使用智能卡的工具。

**先决条件**

- 已安装 opensc 软件包，其中包括 pkcs15-init 工具。

  如需了解更多详细信息，请参阅 安装用于管理和使用智能卡的工具。
- 该卡插入读卡器并连接到计算机。
- 您有一个要存储在智能卡上的私钥、公钥和证书。在此流程中，testuser.key、testuserpublic.key 和 testuser.crt 是用于私钥、公钥和证书的名称。
- 您有当前的智能卡用户 PIN 和 Security Officer PIN(SO-PIN)。

**流程**

1. 擦除智能卡并使用您的 PIN 验证自己：

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

这个卡已经被清除。

2. 初始化智能卡，设置您的用户 PIN 和 PUK，以及您的安全响应 PIN 和 PUK：

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \ --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

pcks15-init 工具在智能卡上创建一个新插槽。

3. 为插槽设置标签和验证 ID：

```
$ pkcs15-init --store-pin --label testuser \ --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

标签设置为人类可读的值，在本例中为 testuser。auth-id 必须是两个十六进制值，在本例中设为 01。

4. 在智能卡的新插槽中存储并标记私钥：

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \ --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```

**注意**

在存储您的私钥并将证书存储在下一步中时，您为 **--id** 指定的值必须相同。建议为 **--id** 指定自己的值，否则它们将更复杂的值由工具计算。

5. 在智能卡上的新插槽中存储并标记该证书：

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \ --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6. 可选：在智能卡的新插槽中保存并标记公钥：

```
$ pkcs15-init --store-public-key testuserpublic.key --label testuserpublic_key --auth-id 01 --id
01 --pin 963214
Using reader with a card: Reader name
```

**注意**

如果公钥与私钥或证书对应，请指定与私钥或证书的 ID 相同的 ID。

7. 可选：某些智能卡要求您通过锁定设置来完成卡：

```
$ pkcs15-init -F
```

此时您的智能卡在新创建的插槽中包含证书、私钥和公钥。您还创建了您的用户 PIN 和 PUK，以及安全响应 PIN 和 PUK。

## 67.7. 在 SSSD.CONF 中配置超时

使用智能卡证书进行身份验证的时间可能比 SSSD 使用的默认超时时间更长。超时时间可能是由以下原因造成的：

- 读取速度慢
- 转发将物理设备组成虚拟环境
- 保存在智能卡中的证书太多
- 如果使用 OCSP 验证证书，则来自 OCSP （在线证书状态协议）响应速度较慢

在这种情况下，您可以将 sssd.conf 文件中的以下超时时间延长为 60 秒：

- p11_child_timeout

- krb5_auth_timeout

**先决条件**

- 您必须以 root 身份登录。

**流程**

1. 打开 sssd.conf 文件：

```
[root@idmclient1 ~]# vim /etc/sssd/sssd.conf
```

2. 更改 p11_child_timeout 的值：

```
[pam]
p11_child_timeout = 60
```

3. 更改 krb5_auth_timeout 的值：

```
[domain/IDM.EXAMPLE.COM]
krb5_auth_timeout = 60
```

4. 保存设置。

现在，在验证失败前，允许与智能卡的交互运行 1 分钟（60 秒）。

## 67.8. 为智能卡身份验证创建证书映射规则

如果要将一个证书用于 AD(Active Directory)和 IdM（身份管理）中的帐户，您可以在 IdM 服务器上创建证书映射规则。

创建此类规则后，用户可以在两个域中使用其智能卡进行身份验证。

有关证书映射规则的详情，请参阅 用于配置身份验证的证书映射规则。

# 第 68 章 在身份管理中配置证书映射规则

证书映射规则是一种便捷的方式，当身份管理(IdM)管理员无法访问某些用户的证书时，用户可以轻松使用证书进行身份验证。这通常是因为证书已由外部证书颁发机构发布。

## 68.1. 用于配置身份验证的证书映射规则

在以下情况下可能需要配置证书映射规则：

- 证书已由 IdM 域处于信任关系的活动目录(AD)的证书系统发布。
- 证书已由外部证书颁发机构发布。
- IdM 环境较大，有很多使用智能卡的用户。在这种情况下，添加完整证书可能会比较复杂。在大多数情况下，主题和签发者是可预测的，因此与完整证书相比，提前添加更容易。

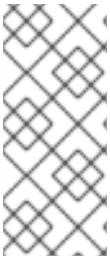
作为系统管理员，您可以创建证书映射规则，并在向特定用户签发证书之前，为用户条目添加证书映射数据。签发证书后，用户可以使用该证书登录，即使完整证书尚未上传到用户条目。

另外，因为证书会定期续订，所以证书映射规则减少了管理开销。续订用户证书时，管理员不必更新用户条目。例如，如果映射基于 Subject 和 Issuer 值，如果新证书的主题和签发者与旧证书相同，则映射仍适用。如果使用完整证书，则管理员必须将新证书上传到用户条目以替换旧证书。

设置证书映射：

1. 管理员必须将证书映射数据或完整证书加载到用户帐户中。
2. 管理员必须创建证书映射规则，以允许其帐户包含与证书上的信息匹配的证书映射数据条目的用户成功登录到 IdM。

创建证书映射规则后，当最终用户提供保存在 文件系统 或 智能卡 上的证书时，身份验证可以成功。



**注意**

密钥分发中心(KDC)有一个证书映射规则的缓存。缓存在第一个 certauth 请求时填充，它有一个硬编码的 300 秒超时。KDC 不会看到对证书映射规则的任何更改，除非它重启了或缓存过期了。

有关构成映射规则的单独组件，以及如何获取和使用它们的详细信息，请参阅 IdM 中的身份映射规则组件，以及获取证书中的签发者，以便在匹配规则中使用。

**注意**

您的证书映射规则可取决于您使用证书的用例。例如，如果您使用带有证书的 SSH，则必须有从证书中提取公钥的完整证书。

## 68.2. IDM 中身份映射规则的组件

您可以在 IdM 中创建 *身份映射规则* 时配置不同的组件。每个组件都有一个可覆盖的默认值。您可以在 Web UI 或 CLI 中定义这些组件。在 CLI 中，身份映射规则使用 ipa certmaprule-add 命令创建。

映射规则

映射规则组件将证书与一个或多个用户帐户关联（或 *映射*）。规则定义将证书与预期用户帐户关联的 LDAP 搜索过滤器。

不同证书颁发机构(CA)发布的证书可能具有不同的属性，并可在不同的域中使用。因此，IdM 不适用于无条件地应用映射规则，而只适用于适当的证书。使用 *匹配规则* 定义适当的证书。

请注意，如果您将映射规则选项留空，则证书将在 userCertificate 属性中作为 DER 编码的二进制文件进行搜索。

利用 --maprule 选项，在 CLI 中定义映射规则。

匹配规则

匹配的规则组件选择您要应用映射规则的证书。默认匹配规则与带有 digitalSignature 密钥 用法和 clientAuth 扩展密钥 用法的证书匹配。

使用 --matchrule 选项，在 CLI 中定义匹配的规则。

域列表

域列表指定您希望 IdM 在处理身份映射规则时搜索用户的身份域。如果您未指定 选项，IdM 仅在 IdM 客户端所属的本地域中搜索用户。

利用 --domain 选项，在 CLI 中定义域。

优先级

当多个规则适用于证书时，优先级最高的规则优先。所有其他规则将被忽略。

- 数字值越低，身份映射规则的优先级越高。例如，具有优先级 1 的规则的优先级高于优先级 2 的规则。
- 如果规则没有定义优先级值，它具有最低的优先级。

使用 --priority 选项，在 CLI 中定义映射规则优先级。

证书映射规则示例

要使用 CLI 定义名为 simple_rule 证书映射规则，如果该证书上的 Subject 与 IdM 中用户帐户中的 certmapdata 条目匹配，则允许对 EXAMPLE.ORG 机构的 智能卡 CA 发布的证书进行身份验证：

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'
```

## 68.3. 从证书获取数据，以便在匹配规则中使用

这个流程描述了如何从证书获取数据，以便您可以将其复制并粘贴到证书映射规则的匹配规则中。要获得匹配规则所需的数据，请使用 sssctl cert-show 或 sssctl cert-eval-rule 命令。

先决条件

- 您有 PEM 格式的用户证书。

流程

1. 创建一个指向证书的变量，该变量还确保其正确编码，以便您可以检索所需的数据。

```
# CERT=$(openssl x509 -in /path/to/certificate -outform der|base64 -w0)
```

2. 使用 sssctl cert-eval-rule 来确定匹配的数据。在以下示例中，使用了证书序列号。

```
# sssctl cert-eval-rule $CERT --match='<ISSUER>CN=adcs19-WIN1-
CA,DC=AD,DC=EXAMPLE,DC=COM' --map='LDAPU1:(altSecurityIdentities=X509:<I>
{issuer_dn!ad_x500}<SR>{serial_number!hex_ur})'
Certificate matches rule.
Mapping filter:

   (altSecurityIdentities=X509:<I>DC=com,DC=example,DC=ad,CN=adcs19-WIN1-
CA<SR>0F0000000000DB8852DD7B246C9C0F0000003B)
```

在这种情况下，将 altSecurityIdentities= 后的所有内容添加到用户的 AD 中的 altSecurityIdentities 属性中。如果使用 SKI 映射，请使用 --map='LDAPU1:(altSecurityIdentities=X509:<SKI>{subject_key_id!hex_u})'。

3. 另外，要根据匹配规则在 CLI 中创建一个新的映射规则，该规则指定证书签发者必须与 ad.example.com 域的 adcs19-WIN1-CA 匹配，证书的序列号必须与用户帐户中的 altSecurityIdentities 条目匹配：

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=adcs19-WIN1-
CA,DC=AD,DC=EXAMPLE,DC=COM' --maprule 'LDAPU1:(altSecurityIdentities=X509:
<I>{issuer_dn!ad_x500}<SR>{serial_number!hex_ur})'
```

## 68.4. 为存储在 IDM 中的用户配置证书映射

如果为其配置的证书身份验证存储在 IdM 中的用户在 IdM 中启用证书映射，系统管理员必须完成以下任务：

- 设置证书映射规则，以便具有与映射规则及其证书映射数据条目中指定的条件匹配的证书的 IdM 用户可以向 IdM 进行身份验证。
- 向 IdM 用户条目输入证书映射数据，以便用户可以使用多个证书进行身份验证，只要它们包含证书映射数据条目中指定的值。

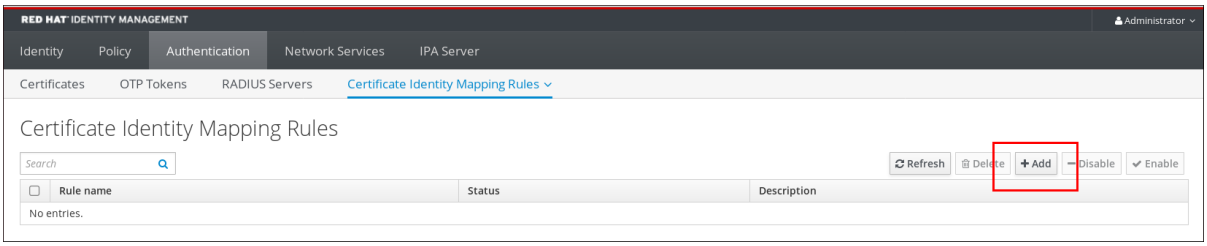
**先决条件**

- 用户在 IdM 中有一个帐户。
- 管理员拥有要添加到用户条目的整个证书或证书映射数据。

### 68.4.1. 在 IdM Web UI 中添加证书映射规则

1. 以管理员身份登录 IdM Web UI。
2. 导航到 **Authentication** → **Certificate Identity Mapping Rules** → **Certificate Identity Mapping Rules**。
3. 点 **Add**。

图 68.1. 在 IdM Web UI 中添加一个新的证书映射规则



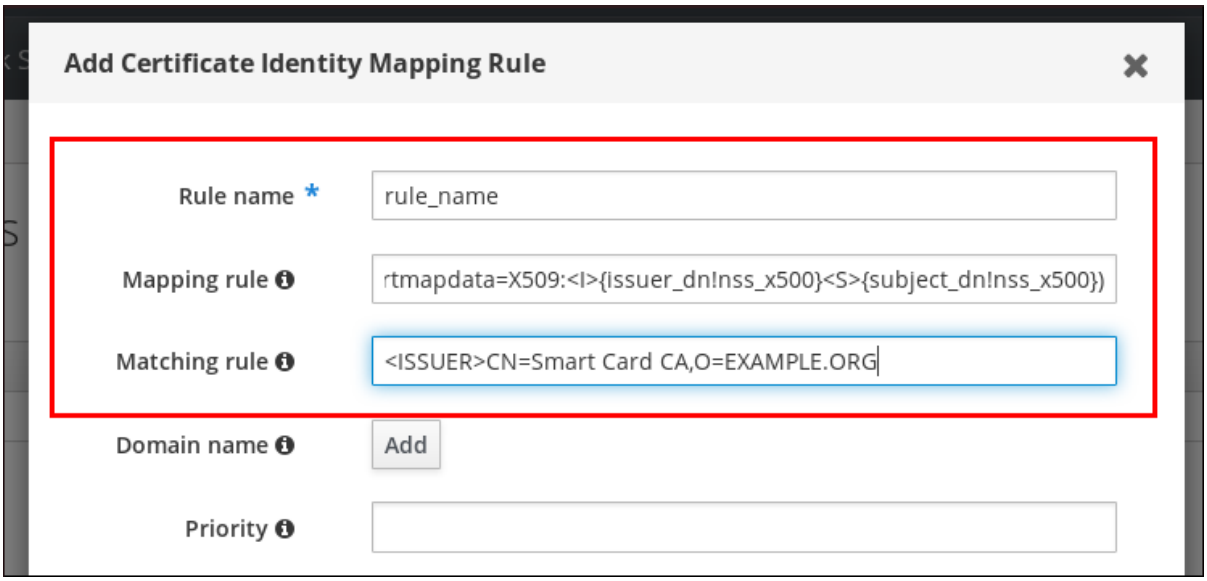

4. 输入规则名称。
5. 输入映射规则。例如，要让 IdM 搜索提供给它们的任何证书中的 Issuer 和 Subject 条目，并根据所显示证书的两个条目中提供的信息做出验证决定：

```
(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})
```

6. 输入匹配的规则。例如，只允许 EXAMPLE.ORG 机构 智能卡 CA 发布的证书来验证用户到 IdM：

```
<ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
```

图 68.2. 在 IdM Web UI 中输入证书映射规则的详情




7. 单击对话框底部的 **Add**，以添加该规则并关闭该框。

8. 系统安全服务守护进程(SSSD)定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

```
# systemctl restart sssd
```

现在，您设置了证书映射规则，可将在智能卡证书中找到的映射规则中指定的数据类型与 IdM 用户条目中的证书映射数据进行比较。找到匹配项后，它将对匹配的用户进行身份验证。

### 68.4.2. 在 IdM CLI 中添加证书映射规则

1. 获取管理员凭证：

```
# kinit admin
```

2. 输入映射规则以及映射规则所基于的匹配规则。例如，要让 IdM 搜索所呈现的任何证书中的发行者和 Subject 条目，并基于所显示证书的两个条目中找到的信息进行身份验证，仅识别由 EXAMPLE.ORG 机构的 智能卡 CA 发布的证书：

```
# ipa certmaprule-add rule_name --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'
----------------------------------------------------
Added Certificate Identity Mapping Rule "rule_name"
----------------------------------------------------
```

```
Rule name: rule_name
Mapping rule: (ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})
Matching rule: <ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
Enabled: TRUE
```

3. 系统安全服务守护进程(SSSD)定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

```
# systemctl restart sssd
```

现在，您设置了证书映射规则，可将在智能卡证书中找到的映射规则中指定的数据类型与 IdM 用户条目中的证书映射数据进行比较。找到匹配项后，它将对匹配的用户进行身份验证。

### 68.4.3. 在 IdM Web UI 中添加证书映射数据到用户条目

1. 以管理员身份登录 IdM Web UI。

2. 导航到 Users → Active users → idm_user。

3. 查找 证书映射数据 选项并单击 Add。

4. 选择以下选项之一：

- 如果您有 idm_user 证书：

  a. 在命令行界面中，使用 cat 工具或文本编辑器显示证书：

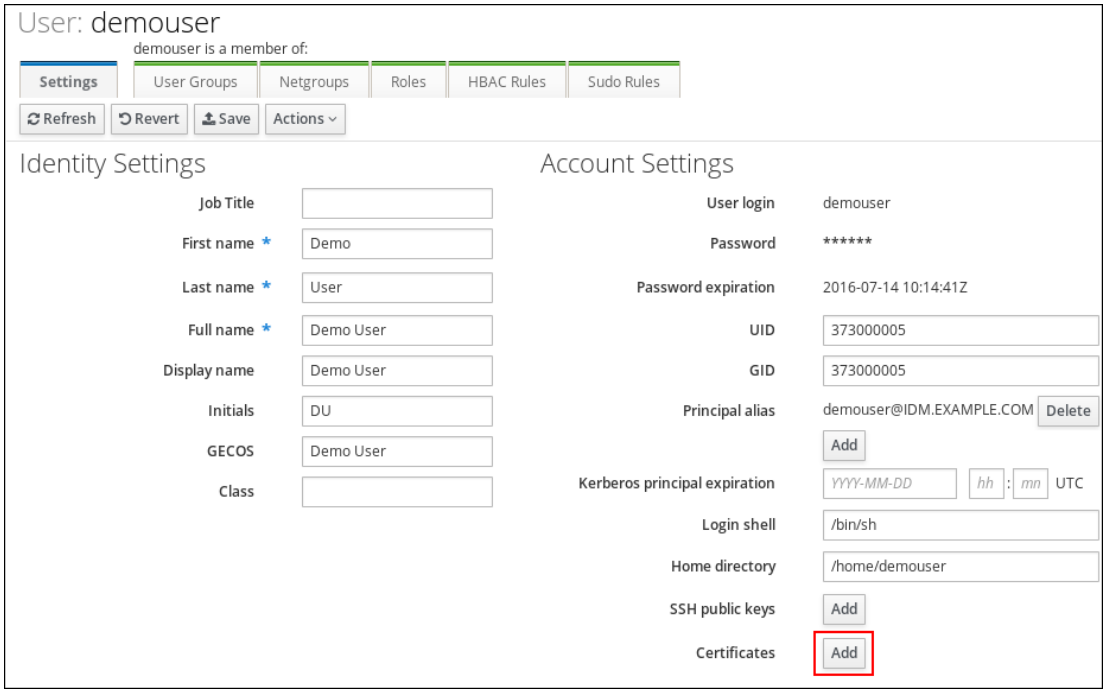
```
[root@server ~]# cat idm_user_certificate.pem
-----BEGIN CERTIFICATE-----
MIIFFTCCA/2gAwIBAgIBEjANBgkqhkiG9w0BAQsFADA6MRgwFgYDVQQKDA9
JRE0u
RVhBTVBMRS5DT00xHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcml0eTAe
Fw0x
ODA5MDIxODE1MzlaFw0yMDA5MDIxODE1MzlaMCwxGDAWBgNVBAoMD0lET
S5FWEFN
[...output truncated...]
```

.

b. **复制**证书。

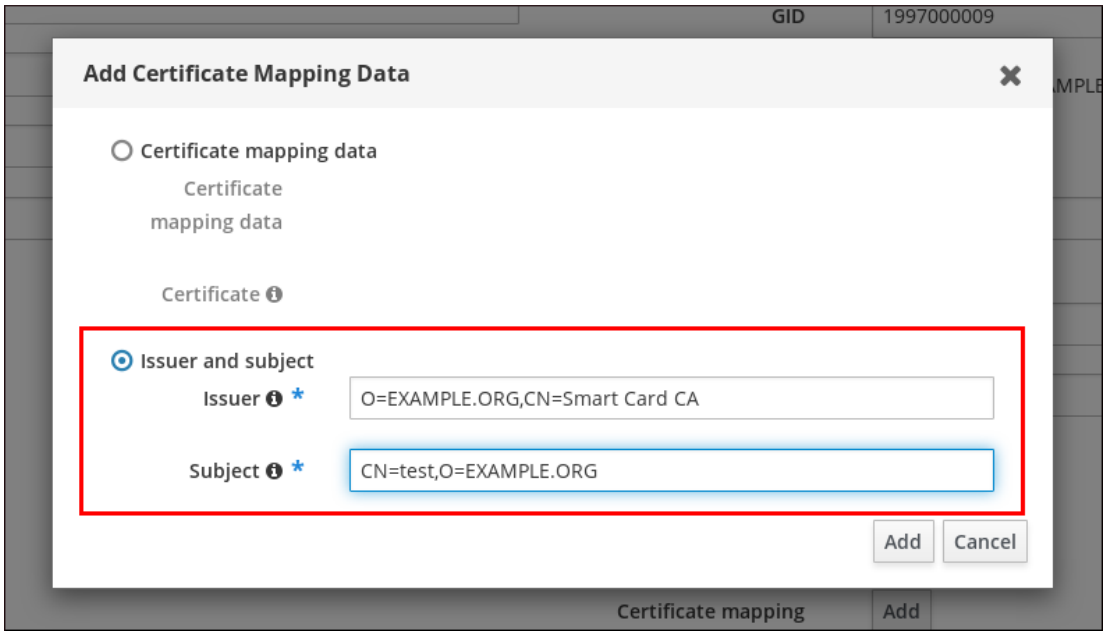
c. **在 IdM Web UI 中，**单击 **Certificate** 旁边的 **Add，并将**证书**粘贴到打开的窗口中。**

**图 68.3. 添加用户证书映射数据：证书**




- **如果您没有 idm_user** 证书，**但知道**证书**的 Issuer 和 Subject，**请检查 **Issuer and subject** 单选按钮，**并在两个框中分**别输入值。

**图 68.4. 添加用户证书映射数据：签发者和主题**



5. 点 Add。

验证步骤

如果您可以访问 .pem 格式的整个证书，请验证是否用户和证书已链接：

1. 使用 sss_cache 程序在 SSSD 缓存中使 idm_user 记录无效，并强制重新载入 idm_user 信息：

```
# sss_cache -u idm_user
```

2. 使用包含 IdM 用户证书的文件名称运行 ipa certmap-match 命令：

```
# ipa certmap-match idm_user_cert.pem
--------------
1 user matched
--------------
 Domain: IDM.EXAMPLE.COM
 User logins: idm_user
----------------------------
Number of entries returned 1
----------------------------
```

输出确认您现在已将证书映射数据添加到 idm_user，并且存在对应的映射规则。这意味着，您可以使用与定义的证书映射数据匹配的任何证书，以 idm_user 进行身份验证。

### 68.4.4. 在 IdM CLI 中添加证书映射数据到用户条目

1. 获取管理员凭证：

```
# kinit admin
```

2. 选择以下选项之一：

- 如果您有 idm_user 证书，请使用 ipa user-add-cert 命令将证书添加到用户帐户中：

```
# CERT=$(openssl x509 -in idm_user_cert.pem -outform der|base64 -w0)
# ipa user-add-certmapdata idm_user --certificate $CERT
```

- 如果您没有 idm_user 证书，但知道用户证书的 Issuer 和 Subject ：

```
# ipa user-add-certmapdata idm_user --subject "O=EXAMPLE.ORG,CN=test" --
issuer "CN=Smart Card CA,O=EXAMPLE.ORG"
-----------------------------------------
Added certificate mappings to user "idm_user"
-----------------------------------------
  User login: idm_user
  Certificate mapping data: X509:<I>O=EXAMPLE.ORG,CN=Smart Card
CA<S>CN=test,O=EXAMPLE.ORG
```

验证步骤

如果您可以访问 .pem 格式的整个证书，请验证是否用户和证书已链接：

1. 使用 sss_cache 程序在 SSSD 缓存中使 idm_user 记录无效，并强制重新载入 idm_user 信息：

```
# sss_cache -u idm_user
```

2. 使用包含 IdM 用户证书的文件名称运行 ipa certmap-match 命令：

```
# ipa certmap-match idm_user_cert.pem
--------------
1 user matched
--------------
 Domain: IDM.EXAMPLE.COM
 User logins: idm_user
----------------------------
Number of entries returned 1
----------------------------
```

输出确认您现在已将证书映射数据添加到 idm_user，并且存在对应的映射规则。这意味着，您可以使用与定义的证书映射数据匹配的任何证书，以 idm_user 进行身份验证。

## 68.5. 使用 ACTIVE DIRECTORY 域信任的证书映射规则

如果 IdM 部署与活动目录(AD)域具有信任关系，则可能会出现不同的证书映射用例。

根据 AD 配置，可能会出现以下情况：

- 如果证书由 AD 证书系统发布，但用户和证书存储在 IdM 中，则身份验证请求的映射和整个处理发生在 IdM 端。有关配置此情境的详情，请参阅为存储在 IdM 中的用户配置证书映射

- 如果用户存储在 AD 中，则身份验证请求的处理会在 AD 中发生。有三个不同的子案例：

  - AD 用户条目包含整个证书。有关在这种情况下配置 IdM 的详情，请参考为 AD 用户条目包含整个证书的用户配置证书映射。

  - AD 配置为将用户证书映射到用户帐户。在本例中，AD 用户条目不包含整个证书，而是包含名为 altSecurityIdentities 的属性。有关如何在这种场景中配置 IdM 的详情，请参阅在将 AD 配置为将用户证书映射到用户帐户时配置证书 映射。

  - AD 用户条目既不包含整个证书，也不包含映射数据。在这种情况下，有两个选项：

    - 如果用户证书是由 AD 证书系统发布的，则证书会包含用户主体名称作为 Subject Alternative Name (SAN)，或者如果最新的更新被应用到 AD，则用户的 SID 在证书的 SID 扩展中。它们都可用于将证书映射到用户。

    - 如果用户证书位于智能卡上，要使用智能卡启用 SSH，SSSD 必须从证书派生公共 SSH 密钥，因此需要完整证书。唯一的解决方案是使用 ipa idoverrideuser-add 命令将整个证书添加到 IdM 中的 AD 用户的 ID 覆盖中。详情请参阅 AD 用户条目不包含证书或映射数据时配置证书映射。

AD 域管理员可以使用 altSecurityIdentities 属性手动将证书映射到 AD 中的用户。此属性支持六个值，但三个映射被视为不安全。作为 2022 年 5 月 10 日安全更新 的一部分，在安装之后，所有设备都处于兼容性模式，如果证书被弱映射到用户，则身份验证如期发生。但是，会记录警告消息，标识任何与完整执行模式不兼容的证书。截止到 2023 年 11 月 14 日，所有设备都将更新为完整的强制模式，如果证书不符合强映射标准，则身份验证将被拒绝。

例如，当 AD 用户请求带有证书(PKINIT)的 IdM Kerberos 票据时，AD 需要在内部将证书映射到用户，并为此使用新的映射规则。但是，在 IdM 中，如果 IdM 用于将证书映射到 IdM 客户端上的用户，则以前的规则将继续工作。

IdM 支持新的映射模板，使 AD 管理员更轻松地使用新规则，而不用维护这两个。IdM 现在支持添加到 Active Directory 的新映射模板，以包括：

-

序列号：LDAPU1: (altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<SR>{serial_number!hex_ur})

- 主题 Key Id: LDAPU1: (altSecurityIdentities=X509:<SKI>{subject_key_id!hex_u})
- 用户 SID：LDAPU1: (objectsid={sid})

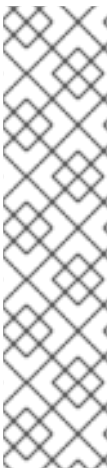
如果您不想使用新的 SID 扩展重新发布证书，您可以通过将适当的映射字符串添加到 AD 中的 altSecurityIdentities 属性来创建手动映射。

### 68.6. 为 AD 用户条目包含整个证书的用户配置证书映射

此用户故事描述了如果 IdM 部署与 Active Directory(AD)信任时，在 IdM 中启用证书映射所需的步骤，该用户存储在 AD 中，AD 中的用户条目包含整个证书。

**先决条件**

- 用户在 IdM 中没有帐户。
- 用户在 AD 中有一个包含证书的帐户。
- IdM 管理员有权访问 IdM 证书映射规则可以基于的数据。



**注意**

为确保 PKINIT 适用于用户，必须应用以下条件之一：

- 用户条目中的证书包括用户的用户主体名称或用户的 SID 扩展。
- AD 中的用户条目在 altSecurityIdentities 属性中有一个合适的条目。

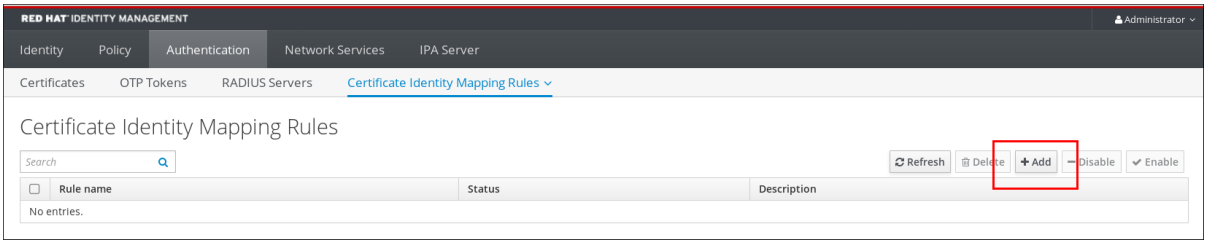
#### 68.6.1. 在 IdM Web UI 中添加证书映射规则

1. 以管理员身份登录 IdM Web UI。

2. 导航到 Authentication → Certificate Identity Mapping Rules → Certificate Identity Mapping Rules。

3. 点 Add。

图 68.5. 在 IdM Web UI 中添加一个新的证书映射规则




4. 输入规则名称。

5. 输入映射规则。与 AD 中的可用内容相比，要向 IdM 提供整个证书以进行身份验证：

```
(userCertificate;binary={cert!bin})
```

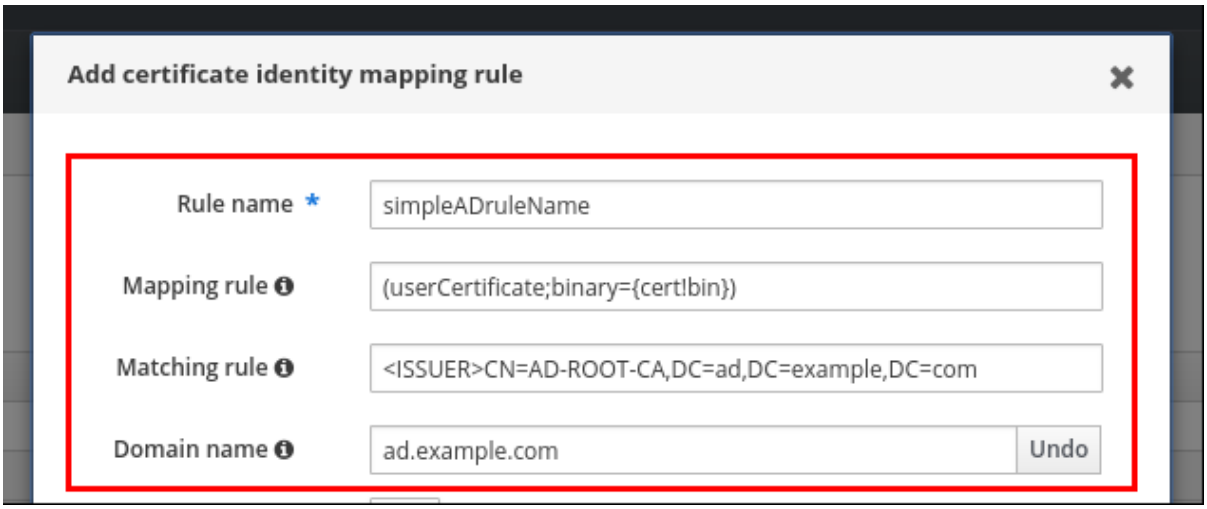
**注意**

如果使用完整证书进行映射，如果续订证书，您必须确保将新证书添加到 AD 用户对象中。

6. 输入匹配的规则。例如，要只允许 AD.EXAMPLE.COM 域的 AD-ROOT-CA 发布的证书进行身份验证：

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

图 68.6. 在 AD 中存储证书的用户的证书映射规则




7. 点 Add。

8. 系统安全服务守护进程(SSSD)定期重新读取证书映射规则。要强制立即加载新创建的规则，请在 CLI 中重启 SSSD：

```
# systemctl restart sssd
```

### 68.6.2. 在 IdM CLI 中添加证书映射规则

1. 获取管理员凭证：

```
# kinit admin
```

2. 输入映射规则以及映射规则所基于的匹配规则。与 AD 中可用的证书相比，要获得进行身份验证的完整证书，只允许 AD.EXAMPLE.COM 域的 AD-ROOT-CA 发布的证书来进行验证：

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-
CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --
domain ad.example.com
----------------------------------------------------
Added Certificate Identity Mapping Rule "simpleADrule"
----------------------------------------------------
  Rule name: simpleADrule
  Mapping rule: (userCertificate;binary={cert!bin})
  Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
  Domain name: ad.example.com
  Enabled: TRUE
```

**注意**

如果使用完整证书进行映射，如果续订证书，您必须确保将新证书添加到 AD 用户对象中。

3. 系统安全服务守护进程(SSSD)定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

```
# systemctl restart sssd
```

## 68.7. 如果将 AD 配置为将用户帐户映射到用户帐户，则配置证书映射

这个用户故事描述了如果 IdM 部署在与活动目录(AD)的信任中，在 IdM 中启用证书映射所需的步骤，用户存储在 AD 中，AD 中的用户条目包含证书映射数据。

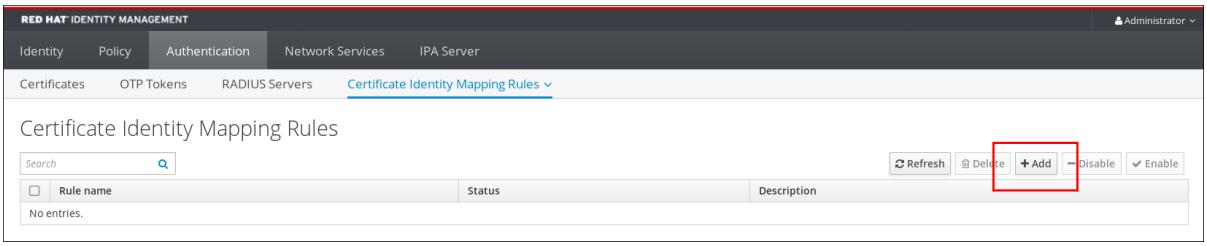
**先决条件**

- 用户在 IdM 中没有帐户。
- 用户在 AD 中有一个帐户，其中包含 altSecurityIdentities 属性，即 IdM certmapdata 属性的 AD 等效。
- IdM 管理员有权访问 IdM 证书映射规则可以基于的数据。

### 68.7.1. 在 IdM Web UI 中添加证书映射规则

1. 以管理员身份登录 IdM Web UI。
2. 导航到 Authentication → Certificate Identity Mapping Rules → Certificate Identity Mapping Rules。
3. 点 Add。

图 68.7. 在 IdM Web UI 中添加一个新的证书映射规则




4. 输入规则名称。

5. 输入映射规则。例如，要使 AD DC 搜索所呈现的任何证书中的 Issuer 和 Subject 条目，并决定根据所出示证书的两个条目中提供的信息进行验证：

```
(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})
```

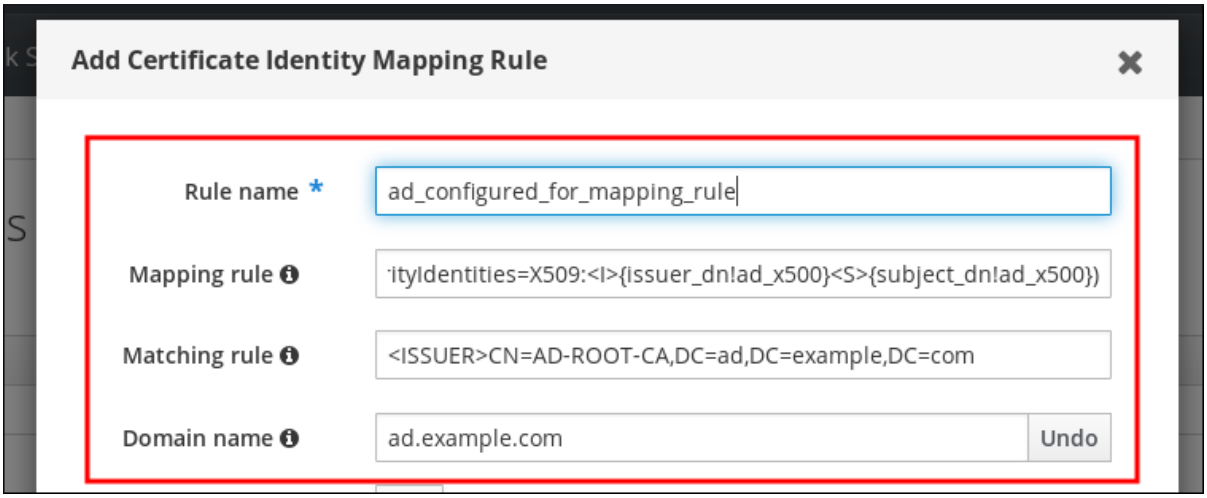
6. 输入匹配的规则。例如，只允许 AD.EXAMPLE.COM 域的 AD-ROOT-CA 发布的证书来将用户认证到 IdM：

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7. 输入域：

```
ad.example.com
```

图 68.8. 如果配置了 AD 进行映射，则证书映射规则




8. 点 Add。

9. 系统安全服务守护进程(SSSD)定期重新读取证书映射规则。要强制立即加载新创建的规则，

请在 CLI 中重启 SSSD：

```
# systemctl restart sssd
```

### 68.7.2. 在 IdM CLI 中添加证书映射规则

1. 获取管理员凭证：

```
# kinit admin
```

2. 输入映射规则以及映射规则所基于的匹配规则。例如，要让 AD 搜索所出示的任何证书中的 Issuer 和 Subject 条目，并且只允许 AD.EXAMPLE.COM 域的 AD-ROOT-CA 发布的证书：

```
# ipa certmaprule-add ad_configured_for_mapping_rule --matchrule
'<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule
'(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})' --
domain=ad.example.com
----------------------------------------------------
Added Certificate Identity Mapping Rule "ad_configured_for_mapping_rule"
----------------------------------------------------
  Rule name: ad_configured_for_mapping_rule
  Mapping rule: (altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>
{subject_dn!ad_x500})
  Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
  Domain name: ad.example.com
  Enabled: TRUE
```

3. 系统安全服务守护进程(SSSD)定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

```
# systemctl restart sssd
```

### 68.7.3. 检查 AD 端的证书映射数据

altSecurityIdentities 属性是与 IdM 中的 certmapdata 用户属性等效的 Active Directory(AD)。当将可信 AD 域配置为将用户帐户映射到用户帐户时，IdM 系统管理员需要检查 AD 中的用户条目是否正确设置了 altSecurityIdentities 属性。

**先决条件**

- 用户帐户必须具有用户管理访问权限。

流程

- 要检查 AD 是否包含 AD 中存储的用户的正确信息，请使用 ldapsearch 命令。例如，输入以下命令检查 adserver.ad.example.com 服务器是否适用以下条件：
  - altSecurityIdentities 属性在 ad_user 的用户条目中设置。
  - matchrule 满足以下条件：
    - ad_user 用于向 AD 进行身份验证的证书由 ad.example.com 域的 AD-ROOT-CA 签发。
    - 主题为 <S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user ：

```
$ ldapsearch -o ldif-wrap=no -LLL -h adserver.ad.example.com \
-p 389 -D cn=Administrator,cn=users,dc=ad,dc=example,dc=com \
-W -b cn=users,dc=ad,dc=example,dc=com "(cn=ad_user)" \
altSecurityIdentities
Enter LDAP Password:
dn: CN=ad_user,CN=Users,DC=ad,DC=example,DC=com
altSecurityIdentities: X509:<I>DC=com,DC=example,DC=ad,CN=AD-ROOT-
CA<S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user
```

## 68.8. 如果 AD 用户条目不包含证书或映射数据，则配置证书映射

如果 IdM 部署与 Active Directory(AD)信任时，此用户故事描述了在 IdM 中启用证书映射所需的步骤，此用户存储在 AD 中，AD 中的用户条目既包含整个证书，也不包含证书映射数据。

先决条件

- 用户在 IdM 中没有帐户。
- 用户在 AD 中有一个帐户，它不包含整个证书和 altSecurityIdentities 属性，即 IdM certmapdata 属性的 AD 等效。
- IdM 管理员已完成了以下任务之一：
  - 将整个 AD 用户证书添加到 IdM 中的 AD 用户的用户 ID 覆盖中

将整个 AD 用户证书添加到 IdM 中的 AD 用户的用户 ID 覆盖 中。

- 创建一个映射到证书中备用字段的证书映射规则，如 Subject Alternative Name 或用户的 SID。

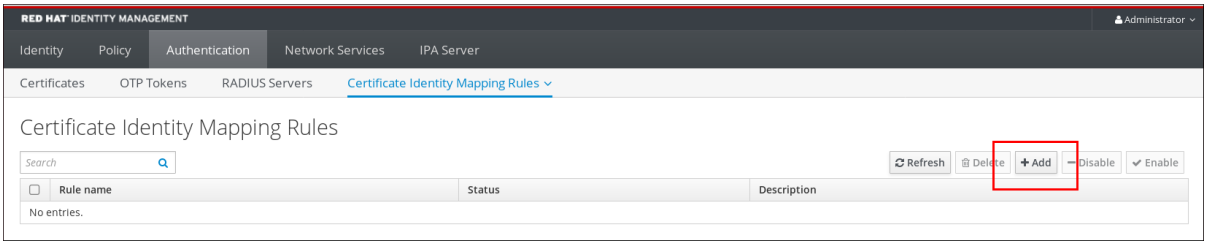
### 68.8.1. 在 IdM Web UI 中添加证书映射规则

1. 以管理员身份登录 IdM Web UI。

2. 导航到 Authentication → Certificate Identity Mapping Rules → Certificate Identity Mapping Rules。

3. 点 Add。

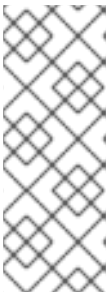
图 68.9. 在 IdM Web UI 中添加一个新的证书映射规则




4. 输入规则名称。

5. 输入映射规则。与存储在 IdM 中的 AD 用户条目的用户 ID 覆盖条目中的证书相比，为 IdM 提供整个证书进行身份验证：

```
(userCertificate;binary={cert!bin})
```



**注意**

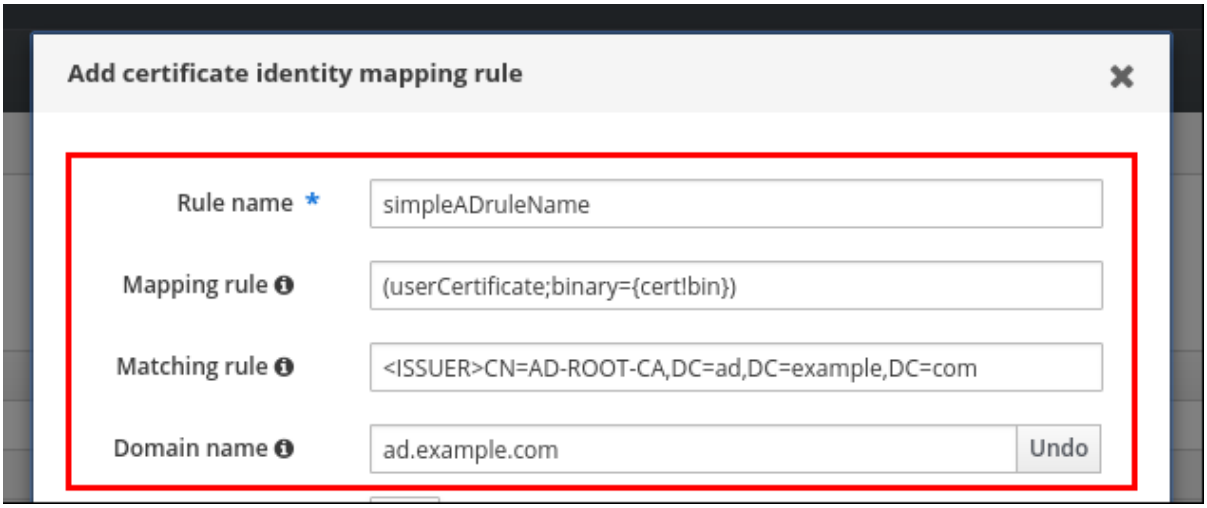
因为证书还包含作为 SAN 或具有最新更新的用户主体名称，所以证书的 SID 扩展中的用户的 SID 也可以使用这些字段将证书映射到用户。例如，如果使用用户的 SID，请将此映射规则替换为 LDAPU1: (objectsid={sid})。有关证书映射的更多信息，请参阅 sss-certmap 手册页。

6. 输入匹配的规则。例如，要只允许 AD.EXAMPLE.COM 域的 AD-ROOT-CA 发布的证书进行身份验证：

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7. 输入域名。例如，要在 ad.example.com 域中搜索用户：

图 68.10. 没有证书或映射数据的用户的证书映射规则




8. 点 Add。

9. 系统安全服务守护进程(SSSD)定期重新读取证书映射规则。要强制立即载入新创建的规则，在 CLI 中重启 SSSD：

```
# systemctl restart sssd
```

### 68.8.2. 在 IdM CLI 中添加证书映射规则

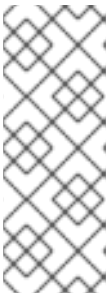
1. 获取管理员凭证：

```
# kinit admin
```

2. 输入映射规则以及映射规则所基于的匹配规则。要让整个证书与存储在 IdM 中的 AD 用户条目的用户 ID 覆盖条目中的证书相比，只允许 AD.EXAMPLE.COM 域的 AD-ROOT-CA 发布的证书来进行验证：

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-
CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --
domain ad.example.com
----------------------------------------------------
Added Certificate Identity Mapping Rule "simpleADrule"
```

```
----------------------------------------------------
Rule name: simpleADrule
Mapping rule: (userCertificate;binary={cert!bin})
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```



**注意**

因为证书还包含作为 SAN 或具有最新更新的用户主体名称，所以证书的 SID 扩展中的用户的 SID 也可以使用这些字段将证书映射到用户。例如，如果使用用户的 SID，请将此映射规则替换为 LDAPU1: (objectsid={sid})。有关证书映射的更多信息，请参阅 sss-certmap 手册页。

3. 系统安全服务守护进程(SSSD)定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

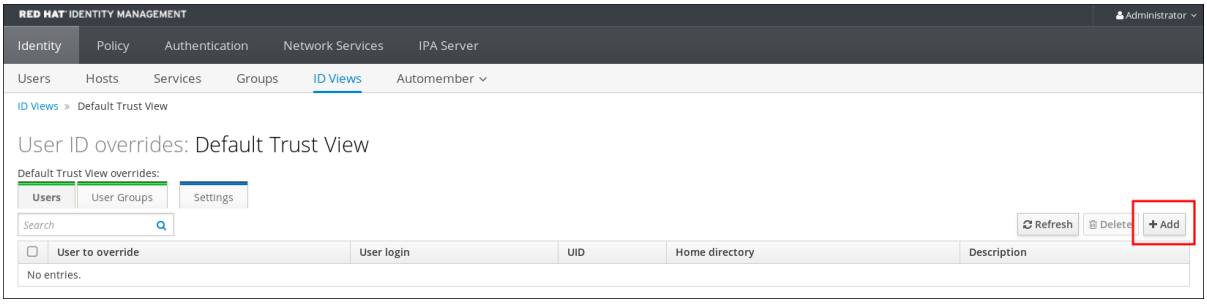
```
# systemctl restart sssd
```

### 68.8.3. 在 IdM Web UI 中添加证书到 AD 用户的 ID 覆盖中

1. 导航到 Identity → ID Views → Default Trust View。

2. 点 Add。

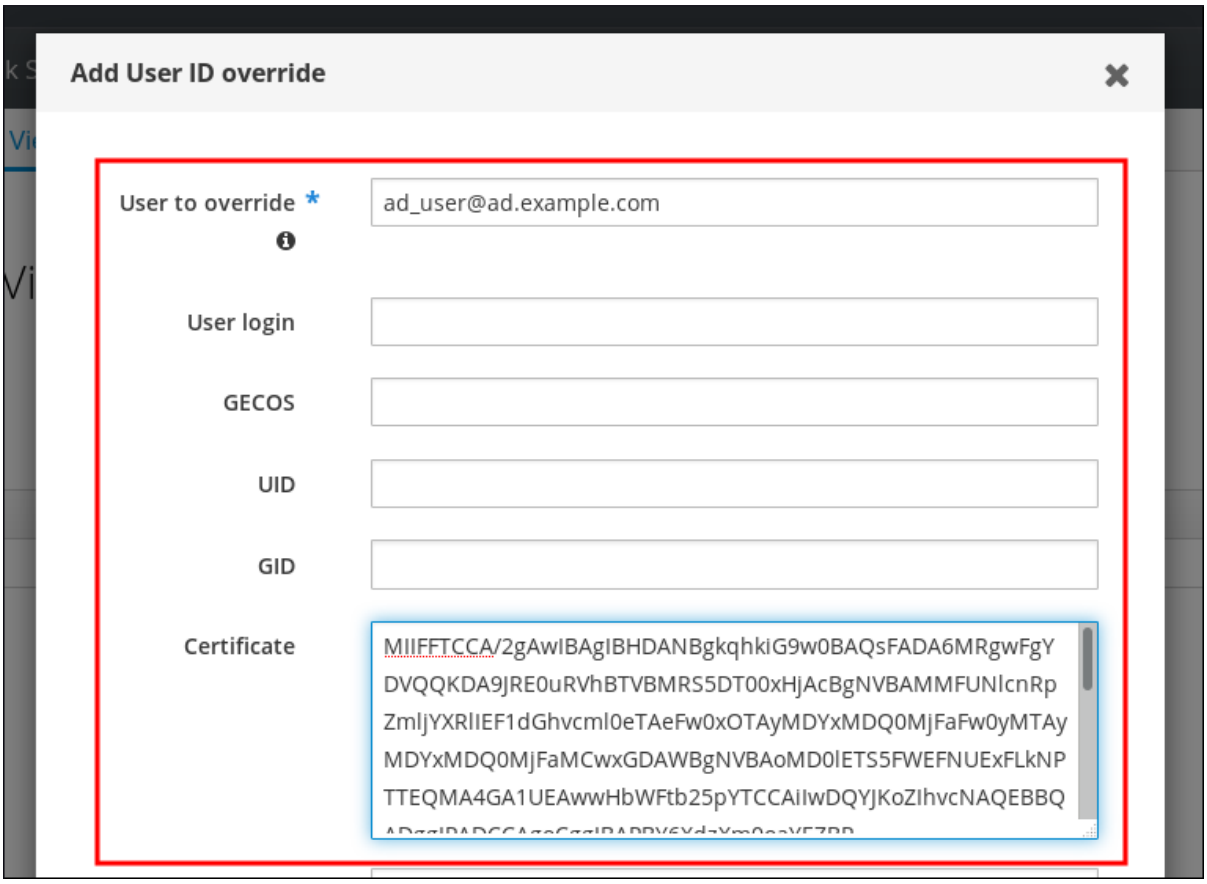
图 68.11. 在 IdM Web UI 中添加一个新的用户 ID 覆盖




3. 在 User to override 字段中，输入 ad_user@ad.example.com。

4. 将 ad_user 的证书复制并粘贴到 Certificate 字段中。

图 68.12. 为 AD 用户配置用户 ID 覆盖




5. 点 Add。

验证步骤

验证用户和证书是否已链接：

1. 使用 sss_cache 程序在 SSSD 缓存中使 ad_user@ad.example.com 记录无效，并强制重新载入 ad_user@ad.example.com 信息：

```
# sss_cache -u ad_user@ad.example.com
```

2. 使用包含 AD 用户证书的文件名称运行 ipa certmap-match 命令：

```
# ipa certmap-match ad_user_cert.pem
--------------
1 user matched
--------------
 Domain: AD.EXAMPLE.COM
 User logins: ad_user@ad.example.com
```

```
----------------------------
Number of entries returned 1
----------------------------
```

输出确认了您已将证书映射数据添加到 ad_user@ad.example.com，并且 如果 AD 用户条目不包含证书或映射数据 中定义的相应的映射规则存在。这意味着，您可以使用与定义的证书映射数据匹配的证书作为 ad_user@ad.example.com 进行身份验证。

其它资源

- 为活动目录用户使用 ID 视图

### 68.8.4. 在 IdM CLI 中在 AD 用户的 ID 覆盖中添加证书

1. 获取管理员凭证：

```
# kinit admin
```

2. 将证书 blob 保存在名为 CERT 的新变量中：

```
# CERT=$(openssl x509 -in /path/to/certificate -outform der|base64 -w0)
```

3. 使用 ipa idoverrideuser-add-cert 命令将 ad_user@ad.example.com 的证书添加到用户帐户中：

```
# ipa idoverrideuser-add-cert ad_user@ad.example.com --certificate $CERT
```

验证步骤

验证用户和证书是否已链接：

1. 使用 sss_cache 程序在 SSSD 缓存中使 ad_user@ad.example.com 记录无效，并强制重新载入 ad_user@ad.example.com 信息：

```
# sss_cache -u ad_user@ad.example.com
```

2. 使用包含 AD 用户证书的文件名称运行 ipa certmap-match 命令：

```
# ipa certmap-match ad_user_cert.pem
--------------
1 user matched
--------------
 Domain: AD.EXAMPLE.COM
 User logins: ad_user@ad.example.com
----------------------------
Number of entries returned 1
----------------------------
```

输出确认了您已将证书映射数据添加到 ad_user@ad.example.com，并且 如果 AD 用户条目不包含证书或映射数据 中定义的相应的映射规则存在。这意味着，您可以使用与定义的证书映射数据匹配的证书作为 ad_user@ad.example.com 进行身份验证。

其它资源

- 为活动目录用户使用 ID 视图

## 68.9. 将多个身份映射规则合并到一个规则中

要将多个身份映射规则组合成一个组合规则，请使用 | （或）字符在单个映射规则前前，并使用 （） 方括号将它们分隔，例如：

证书映射过滤器示例 1

```
$ ipa certmaprule-add ad_cert_for_ipa_and_ad_users \ --maprule='(|(ipacertmapdata=X509:<I>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})(altSecurityIdentities=X509:<I>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \ --matchrule='<ISSUER>CN=AD-ROOT-
CA,DC=ad,DC=example,DC=com' \ --domain=ad.example.com
```

在上例中，-- maprule 选项中的过滤器定义包括这些条件：

- ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500} 是一个过滤器，它将一个智能卡证书中的 subject 和 issuer 连接到一个 IdM 用户账户中的 ipacertmapdata 属性的值，如 Adding a certificate mapping rule in IdM 部分所述
- altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500} 是一个过

滤器，它将一个智能卡证书中的 subject 和 issuer 连接到一个 AD 用户账户中的 altSecurityIdentities 属性的值，如 Adding a certificate mapping rule if the trusted AD domain is configured to map user certificates 所述

- 添加 --domain=ad.example.com 选项意味着映射到给定证书的用户不仅在本地 idm.example.com 域中搜索，还要在 ad.example.com 域中搜索

--maprule 选项中的过滤器定义接受逻辑运算符 |（或），以便您可以指定多个条件。在这种情况下，规则映射了至少满足其中一个条件的所有用户帐户。

证书映射过滤器示例 2

```
$ ipa certmaprule-add ipa_cert_for_ad_users \
  --maprule='(|(userCertificate;binary={cert!bin})(ipacertmapdata=X509:<I>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})(altSecurityIdentities=X509:<I>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \
  --matchrule='<ISSUER>CN=Certificate Authority,O=REALM.EXAMPLE.COM' \
  --domain=idm.example.com --domain=ad.example.com
```

在上例中，-- maprule 选项中的过滤器定义包括这些条件：

- userCertificate;binary={cert!bin} 是一个过滤器，它返回包含整个证书的用户条目。对于 AD 用户，创建这类过滤器在 如果 AD 用户条目不包含证书或映射数据，请添加一个证书映射规则 中进行了详细描述。

- ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500} 是一个过滤器，它将一个智能卡证书中的 subject 和 issuer 连接到一个 IdM 用户账户中的 ipacertmapdata 属性的值，如 Adding a certificate mapping rule in IdM 部分所述。

- altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500} 是一个过滤器，它将一个智能卡证书中的 subject 和 issuer 连接到一个 AD 用户账户中的 altSecurityIdentities 属性的值，如 Adding a certificate mapping rule if the trusted AD domain is configured to map user certificates 所述。

--maprule 选项中的过滤器定义接受逻辑运算符 |（或），以便您可以指定多个条件。在这种情况下，规则映射了至少满足其中一个条件的所有用户帐户。

### 68.10. 其它资源

- 请参阅 **sss-certmap(5)** 手册页。

# 第 69 章 使用存储在 IDM 客户端桌面的证书配置身份验证

通过配置身份管理(IdM)，IdM 系统管理员可以使用向用户签发的证书颁发机构(CA)的证书，使用户能够向 IdM Web UI 和命令行界面(CLI)进行身份验证。证书存储在 IdM 客户端的桌面上。

Web 浏览器可以在不属于 IdM 域的系统中运行。

在使用证书配置身份验证时请注意以下几点：

- 如果您要使用证书进行身份验证的用户已有证书，则您可以跳过 请求新的用户证书并将其导出到客户端 ；
- 如果用户的证书已由 IdM CA 发布了，则您可以跳过 确保证书和用户链接在一起。

**注意**

只有身份管理用户可以使用证书登录 Web UI。Active Directory 用户可使用其用户名和密码登录。

## 69.1. 在 WEB UI 中为证书验证配置身份管理服务器

作为身份管理(IdM)管理员，您可以允许用户使用证书为您的 IdM 环境进行身份验证。

**流程**

作为身份管理管理员：

1. 在身份管理服务器上，获取管理员特权并创建 shell 脚本来配置服务器。
   a. 运行 ipa-advise config-server-for-smart-card-auth 命令，并将其输出保存到文件中，如 server_certificate_script.sh ：

```
# kinit admin
# ipa-advise config-server-for-smart-card-auth > server_certificate_script.sh
```

b. 使用 chmod 实用程序为文件添加执行权限：

```
# chmod +x server_certificate_script.sh
```

2. 在 Identity Management 域中的所有服务器上，运行 server_certificate_script.sh 脚本

a. 使用 IdM 证书颁发机构证书的路径 /etc/ipa/ca.crt，因为如果 IdM CA 是唯一签发了您要为其启用证书验证的用户证书的证书颁发机构：

```
# ./server_certificate_script.sh /etc/ipa/ca.crt
```

b. 如果不同的外部 CA 签署您想要为其启用证书验证的用户证书，则使用路径作为输入：

```
# ./server_certificate_script.sh /tmp/ca1.pem /tmp/ca2.pem
```

注意

如果要为整个拓扑中启用用户的证书身份验证，请不要忘记在将来添加到系统的每个新副本上运行 脚本。

## 69.2. 请求新的用户证书并将其导出到客户端

作为身份管理(IdM)管理员，您可以为 IdM 环境中的用户创建证书，并将其导出到您要为用户启用证书身份验证的 IdM 客户端。

注意

如果要使用证书进行身份验证的用户已有证书，则不需要按照以下流程操作。

**流程**

1. （可选）创建新目录，如 ~/certdb/，并使其成为临时证书数据库。当系统提示时，创建一个 NSS 证书数据库密码来加密后续步骤中生成的证书的密钥：

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
```

```
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2. 创建证书签名请求(CSR)，并将输出重定向到文件。例如，要为 IDM.EXAMPLE.COM 域中的 idm_user 用户创建一个名称为 certificate_request.csr 的 4096 位 CSR，请将证书私钥的昵称设为 idm_user 以便于查找，并将主题设为 CN=idm_user,O=IDM.EXAMPLE.COM ：

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s
"CN=idm_user,O=IDM.EXAMPLE.COM" > certificate_request.csr
```

3. 出现提示时，输入您在使用 certutil 创建临时数据库时输入的相同密码。然后继续键入 rundlomly 直到通知停止：

```
Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the
creation of your key.  One of the easiest ways to create a
random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter
is full.  DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!


Continue typing until the progress meter is full:
```

4. 将证书请求文件提交到服务器。指定要与新发布的证书关联的 Kerberos 主体、存储证书的输出文件，以及可选的证书配置集。例如，要获取 IECUserRoles 配置集的证书，带有添加的用户角色扩展的配置文件，idm_user@IDM.EXAMPLE.COM 主体，并将它保存在 ~/idm_user.pem 文件中：

```
# ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --certificate-out=~/idm_user.pem
```

5. 将证书添加到 NSS 数据库。使用 -n 选项设置之前创建 CSR 时所用的相同 nickname，以便该证书与 NSS 数据库中的私钥相匹配。t 选项设置信任级别。详情请查看 certutil(1)man page。i 选项指定 输入证书文件。例如，要将一个带有 idm_user 昵称的证书添加到 NSS 数据库中，该证书存储在~/certdb/ 数据库的 ~/idm_user.pem 文件中：

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

6. 验证 NSS 数据库中的密钥是否未显示 （或称为） 作为其 nickname。例如，验证存储在 ~/certdb/ 数据库中 的证书没有被孤立：

```
# certutil -K -d ~/certdb/
< 0> rsa      5ad14d41463b87a095b1896cf0068ccc467df395   NSS Certificate
DB:idm_user
```

7. 使用 pk12util 命令将证书从 NSS 数据库导出到 PKCS12 格式。例如，将 /root/certdb NSS 数据库中的 idm_user nickname 的证书导出到 ~/idm_user.p12 文件：

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

8. 将证书传输到您要启用 idm_user 的证书身份验证的主机：

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

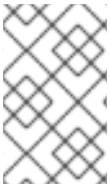
9. 在传输了证书的主机上，出于安全原因使 'pkcs12 文件被 'other' 组无法访问的目录：

```
# chmod o-rwx /home/idm_user/
```

10. 出于安全考虑，请从服务器中删除临时 NSS 数据库和 .pkcs12 文件：

```
# rm ~/certdb/
# rm ~/idm_user.p12
```

### 69.3. 确保证书和用户链接在一起



**注意**

如果用户的证书已由 IdM CA 发布，则不需要按照此流程操作。

要使证书身份验证发挥作用，您需要确保证书链接到将使用证书进行身份管理(IdM)身份验证的用户。

- 如果证书是由不属于您的身份管理环境的证书颁发机构提供的，请根据 将用户帐户链接到证书 中描述的流程链接用户和证书。

- 如果证书由 Identity Management CA 提供，则证书会自动添加到用户条目中，您不必将该证书链接到用户帐户。有关在 IdM 中创建新证书的详情，请参阅 请求新的用户证书并将其导出到客户端。

## 69.4. 配置浏览器以启用证书身份验证

若要在使用 Web UI 登录身份管理(IdM)时通过证书进行身份验证，您需要将用户和相关证书颁发机构(CA)证书导入到 Mozilla Firefox 或 Google Chrome 浏览器。浏览器运行的主机本身不必是 IdM 域的一部分。

IdM 支持以下浏览器来连接到 WebUI：

- Mozilla Firefox 38 及更新的版本
- Google Chrome 46 及更新的版本

以下流程演示了如何配置 Mozilla Firefox 57.0.1 浏览器。

### 先决条件

- 您可以使用 PKCS#12 格式在浏览器中导入 用户证书。

### 流程

1. 打开 Firefox，然后导航到 Preferences → Privacy & Security。

图 69.1. Preferences 中的隐私和安全部分

2. 单击 查看证书。

图 69.2. 查看隐私和安全性中的证书

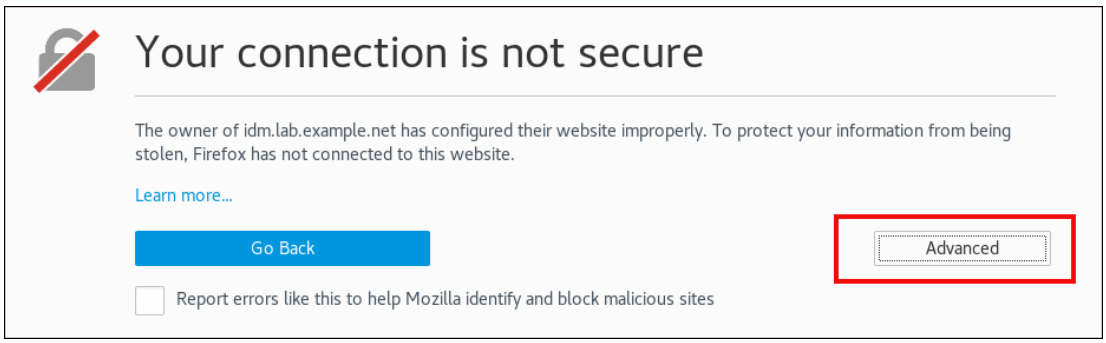

3. 在您的 证书 选项卡中，单击 Import。以 PKCS12 格式查找并打开用户证书，然后点 OK 和 OK。

4. 确保 Identity Management 证书授权机构被 Firefox 认可为可信颁发机构：

a. 在本地保存 IdM CA 证书：

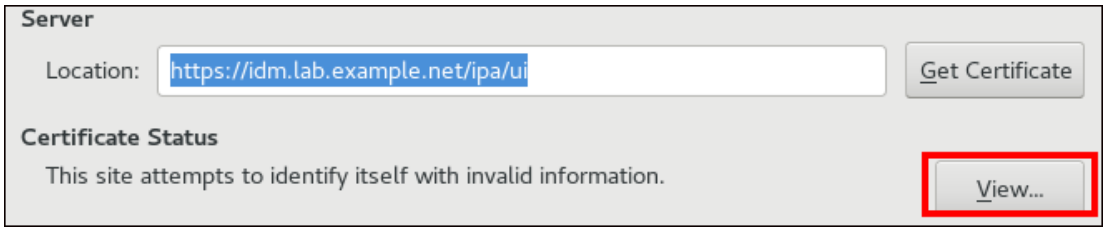
- 通过在 Firefox 地址栏中写入 IdM 服务器的名称，导航到 IdM Web UI。在 Insecure Connection 警告页面上单击 Advanced。

图 69.3. 不安全的连接



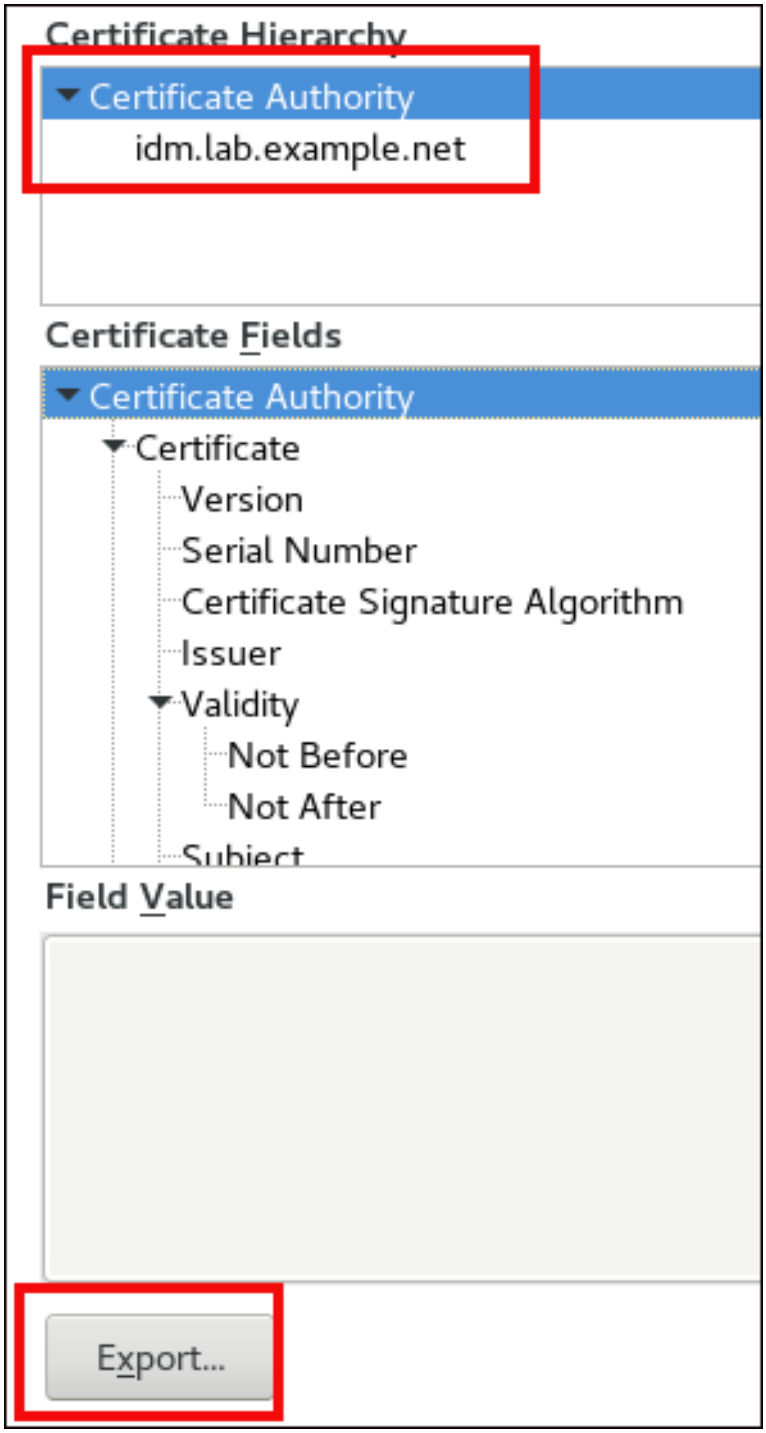

- 添加例外.单击 View。

图 69.4. 查看证书的详情



- 在 详细信息 选项卡中，突出显示 证书颁发机构 字段。

图 69.5. 导出 CA 证书




- 单击 Export。保存 CA 证书，如 CertificateAuthority.crt 文件，然后单击 Close 和 Cancel。

b. 将 IdM CA 证书导入 Firefox 作为可信证书颁发机构证书：

- 打开 Firefox，导航到 Preferences 并点击 Privacy & Security。

图 69.6. Preferences 中的隐私和安全部分


- 单击 查看证书。

图 69.7. 查看隐私和安全性中的证书


- 在" 颁发机构 "选项卡中，单击 Import。查找并打开您在上一步中在 CertificateAuthority.crt 文件中保存的 CA 证书。信任证书来识别网站，然后点OK 和 OK。

5. 继续 验证身份管理 Web UI，并使用作为身份管理用户的证书。

## 69.5. 以身份管理用户的身份使用证书向身份管理 WEB UI 进行身份验证

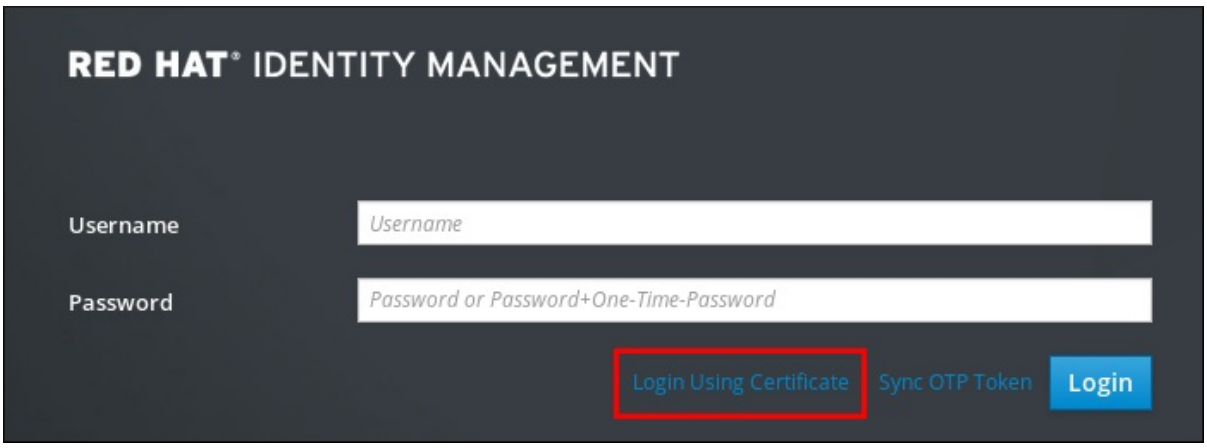
按照以下流程，使用存储在身份管理客户端桌面上的证书作为用户向身份管理(IdM) Web UI 进行身份验证。

**流程**

1. 在浏览器中，导航到位于 的 Identity Management Web UI，例如 https://server.idm.example.com/ipa/ui。

2. 单击 "登录使用证书 "

图 69.8. 在身份管理 Web UI 中 使用证书登录




3. 应该已经选择了用户的证书。取消选中 Remember this Decision，然后单击 OK。

现在，您被验证为与证书对应的用户。

**其它资源**

- 请参阅 为智能卡验证配置身份管理。

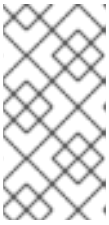
## 69.6. 配置 IDM 客户端以使用证书启用对 CLI 的身份验证

要使 IdM 用户在 IdM 客户端的命令行界面(CLI)中为 IdM 用户提供证书身份验证，请将 IdM 用户的证书和私钥导入到 IdM 客户端。有关创建和传输用户证书的详情，请参阅 请求新的用户证书并将其导出到客户端。

**流程**

- 登录 IdM 客户端，并让包含用户证书和私钥的 .p12 文件准备好。要获取并缓存 Kerberos 票据授予票据(TGT)，请使用带有用户主体的 -X 选项和 X509_username:/path/to/file.p12 属性运行 kinit 命令，以指定在何处查找用户的 X509 身份信息。例如，要使用存储在 ~/idm_user.p12 文件中的用户身份信息获取 idm_user 的 TGT：

```
$ kinit -X X509_idm_user='PKCS12:~/idm_user.p12' idm_user
```



**注意**

该命令还支持 .pem 文件格式： **kinit -X X509_username='FILE:/path/to/cert.pem,/path/to/key' user_principal**

# 第 70 章 使用 IDM CA 续订服务器

## 70.1. IDM CA 续订服务器解释

在使用嵌入式证书颁发机构 (CA) 的 Identity Management (IdM) 部署中，CA 续订服务器维护并更新 IdM 系统证书。它确保了强大的 IdM 部署。

IdM 系统证书包括：

- IdM CA 证书
- OCSP 签名证书
- IdM CA 子系统 证书
- IdM CA 审计签名 证书
- IdM 续订代理 (RA)证书
- KRA 传输和存储证书

对系统证书进行定性的特征是，它们的密钥由所有 CA 副本共享。相比之下，IdM 服务证书（如 LDAP、HTTP 和 PKINIT 证书）在不同的 IdM CA 服务器上具有不同的密钥对和主题名称。

在 IdM 拓扑中，默认情况下，第一个 IdM CA 服务器是 CA 续订服务器。

**注意**

在上游文档中，IdM CA 称为 Dogtag。

### CA 续订服务器的角色

IdM CA、IdM CA 子系统 和 IdM RA 证书对 IdM 部署至关重要。每个证书都存储在 /etc/pki/pki-tomcat/ 目录中的 NSS 数据库以及 LDAP 数据库条目中。存储在 LDAP 中的证书必须与存储在 NSS 数据库中的证书匹配。如果不匹配，在 IdM 框架和 IdM CA 之间以及 IdM CA 和 LDAP 之间会发生身份验证失败。

所有 IdM CA 副本都有针对每个系统证书的跟踪请求。如果带有集成 CA 的 IdM 部署不包含 CA 续订服务器，则每个 IdM CA 服务器都会单独请求续订系统证书。这会导致发生各种系统证书和身份验证失败的不同 CA 副本。

将一个 CA 副本用作续订服务器，可以在需要时完全续订一次系统证书，从而避免身份验证失败。

### CA 副本上的 cert monger 服务的角色

在所有 IdM CA 副本上运行的 certmonger 服务 使用 dogtag-ipa-ca-renew-agent 续订帮助程序来跟踪 IdM 系统证书。续订帮助程序读取 CA 续订服务器配置。在不是 CA 续订服务器的每个 CA 副本上，续订帮助程序从 ca_renewal LDAP 条目检索最新的系统证书。由于正好发生 证书续订 尝试时，dog tag-ipa-ca-renew-agent 帮助程序有时会在 CA 续订服务器实际续订证书前尝试更新系统证书。如果发生这种情况，旧的、即将扩展的证书将返回到 CA 副本上的 certmonger 服务。在意识到它 的证书服务 已存储在其数据库中，证书服务会一直尝试在单独尝试之间延迟更新证书，直到它可以从 CA 续订服务器检索更新的证书。

### IdM CA 续订服务器正常工作

带有嵌入式 CA 的 IdM 部署是一个 IdM 部署，安装有 IdM CA - 或者稍后安装了 IdM CA 服务器。带有嵌入式 CA 的 IdM 部署必须始终有一个 CA 副本配置为续订服务器。续订服务器必须在线且功能完整，并且必须与其他服务器正确复制。

如果要使用 ipa server-del、ipa-replica-manage del、ipa-csreplica-manage del 或 ipa-server-install --uninstall 命令删除当前的 CA 续订服务器，则另一个 CA 副本会自动分配为 CA 续订服务器。此策略确保续订服务器配置保持有效。

该政策不包括以下情况：

- 脱机续订服务器

  如果续订服务器在延长期限内处于脱机状态，则可能会错过续订窗口。在这种情况下，所有非续订 CA 服务器都会持续重新安装当前的系统证书，直到证书过期为止。当发生这种情况时，IdM 部署会被破坏，因为即使是一个过期的证书都可能会导致其他证书的续订失败。

  为防止这种情况：如果您当前的续订服务器离线且长时间不可用，请考虑 手动分配新的 CA

续订服务器。

- 复制问题

  如果在续订服务器和其他 CA 副本之间存在复制问题，则续订可能会成功，但其他 CA 副本可能无法在更新的证书过期前检索更新的证书。

  要防止这种情况，请确保您的复制协议正常工作。详情请参阅 RHEL 7 *Linux 域身份、身份验证和策略指南*中的常规或特定复制故障排除指南。

## 70.2. 更改和重置 IDM CA 续订服务器

当证书颁发机构(CA)续订服务器停用时，身份管理(IdM)会自动从 IdM CA 服务器列表中选择一个新的 CA 续订服务器。系统管理员无法影响选择。

为了能够选择新的 IdM CA 续订服务器，系统管理员必须手动执行替换操作。在开始停用当前续订服务器的过程之前，选择新的 CA 续订服务器。

如果当前的 CA 续订服务器配置无效，请重置 IdM CA 续订服务器。

完成此步骤以更改或重置 CA 续订服务器。

**先决条件**

- 有 IdM 管理员凭证。

**流程**

1. 获取 IdM 管理员凭证：

```
~]$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

2. （可选）查找部署中的哪些 IdM 服务器具有必要的 CA 角色，有资格成为新的 CA 续订服务器：

```
~]$ ipa server-role-find --role 'CA server'
---------------------
2 server roles matched
---------------------
  Server name: server.idm.example.com
  Role name: CA server
  Role status: enabled

  Server name: replica.idm.example.com
  Role name: CA server
  Role status: enabled
----------------------------
Number of entries returned 2
----------------------------
```

部署中有两个 CA 服务器。

3. 另外，要查找哪个 CA 服务器是当前 CA 续订服务器，请输入：

```
~]$ ipa config-show | grep 'CA renewal'
  IPA CA renewal master: server.idm.example.com
```

当前续订服务器为 server.idm.example.com。

4. 要更改续订服务器配置，请使用 ipa config-mod 实用程序和 --ca-renewal-master-server 选项：

```
~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com | grep 'CA
renewal'
  IPA CA renewal master: replica.idm.example.com
```

重要

您还可以使用以下命令切换到新的 CA 续订服务器：

- ipa-cacert-manage --renew 命令。此命令会续订 CA 证书，*并使* 您在其上执行新 CA 续订服务器的 CA 服务器。

- ipa-cert-fix 命令。当证书过期时，该命令会恢复部署。它还使您在其上执行该命令的 CA 服务器成为新的 CA 续订服务器。

详情请查看 IdM 离线时重新更新过期的系统证书。

# 第 71 章 管理外部签名的 CA 证书

身份管理(IdM)提供不同类型的证书颁发机构(CA)配置。您可以选择安装带有集成 CA 或带有外部 CA 的 IdM。您必须指定在安装过程中使用的 CA 类型。但是，安装后，您可以从外部签名的 CA 移到自签名 CA，反之亦然。另外，当自动续订自签名 CA 时，您必须确保续订外部签名的 CA 证书。请参考管理外部签名的 CA 证书所需的相关部分。

- 安装带有外部签名 CA 的 IdM：
  - 安装带有集成 DNS 和外部 CA 作为根 CA 的 IdM 服务器。
  - 安装没有集成 DNS 和外部 CA 作为根 CA 的 IdM 服务器。
- 从外部签名的 CA 切换到自签名 CA。
- 从自签名 CA 切换到外部签名的 CA。
- 续订外部签名的 CA 证书。

## 71.1. 在 IDM 中从外部签名的 CA 切换到自签名 CA

完成此步骤，从外部签名切换到身份管理(IdM)证书认证机构(CA)的自签名证书。使用自签名 CA 时，自动管理 CA 证书的续订：系统管理员不需要向外部机构提交证书签名请求(CSR)。

从外部签名切换到自签名 CA 只替换 CA 证书。之前 CA 签名的证书仍有效且仍在使用。例如，即使您移至自签名 CA 后，LDAP 证书的证书链也会保持不变：

```
external_CA certificate > IdM CA certificate > LDAP certificate
```

**先决条件**

- 您有访问 IdM CA 续订服务器和所有 IdM 客户端及服务器的 root 权限。

**流程**

1. 在 IdM CA 续订服务器上，将 CA 证书更新为自签名：

```
# ipa-cacert-manage renew --self-signed
Renewing CA certificate, please wait
CA certificate successfully renewed
The ipa-cacert-manage command was successful
```

2. 以 root 身份 SSH 到所有剩余的 IdM 服务器和客户端。例如：

```
# ssh root@idmclient01.idm.example.com
```

3. 在 IdM 客户端上，使用来自服务器的证书更新本地 IdM 证书数据库：

```
[idmclient01 ~]# ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

4. （可选）检查您的更新是否成功，并将新 CA 证书添加到 /etc/ipa/ca.crt 文件中：

```
[idmclient01 ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -
print_certs -text -noout
[...]
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 39 (0x27)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority
        Validity
            Not Before: Jul  1 16:32:45 2019 GMT
            Not After : Jul  1 16:32:45 2039 GMT
        Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority
[...]
```

输出显示更新已成功，因为新 CA 证书使用旧的 CA 证书列出。

## 71.2. 在 IDM 中从自签名 CA 切换到外部签名的 CA

在 IdM 中，您可以从自签名 CA 切换到外部签名的 CA。在 IdM 中切换到外部签名的 CA 后，您的 IdM CA 服务器变为外部 CA 的子 CA。另外，不会自动管理 CA 证书的续订，系统管理员必须向外部颁发机构提交证书签名请求(CSR)。

要切换到外部签名的 CA，CSR 必须由外部 CA 签名。按照 使用外部 CA 续订 IdM CA 续订服务器证书 中的步骤操作，来切换到 IdM 中的自签名 CA。

### 71.3. 使用外部 CA 续订 IDM CA 续订服务器证书

按照以下流程，使用外部 CA 续订身份管理(IdM)证书颁发机构(CA)证书，来签名证书签名请求(CSR)。在这个配置中，您的 IdM CA 服务器是外部 CA 的 subCA。外部 CA 可以（但不必）是 Active Directory 证书服务器(AD CS)。

如果外部证书颁发机构是 AD CS，您可以在 CSR 中为 IdM CA 证书指定您想要的模板。证书模板定义收到证书请求时使用的策略和规则。AD 中的证书模板与 IdM 中的证书配置集对应。

您可以通过其对象标识符(OID)定义特定的 AD CS 模板。OID 是不同发布机构发布的唯一数字值，用于唯一标识分布式应用中的数据元素、语法和其他部分。

另外，您还可以根据名称来定义特定的 AD CS 模板。例如，IdM CA 向 AD CS 提交的 CSR 中使用的默认配置集的名称是 subCA。

要通过在 CSR 中指定 OID 或名称来定义配置集，请使用 external-ca-profile 选项。详情请查看 ipa-cacert-manage man page。

除了使用现成的证书模板外，您还可以在 AD CS 中创建自定义证书模板，并在 CSR 中使用它。

**先决条件**

- 具有 IdM CA 续订服务器的 root 访问权限。

**流程**

完成此流程以使用外部签名续订 IdM CA 的证书，无论当前的 CA 证书是自签名还是外部签名。

1. 创建要提交到外部 CA 的 CSR：

   - 如果外部 CA 是 AD CS，请使用 --external-ca-type=ms-cs 选项。如果要使用与默认 subCA 模板不同的模板，请使用 --external-ca-profile 选项指定它：

```
~]# ipa-cacert-manage renew --external-ca --external-ca-type=ms-cs [--external-ca-profile=PROFILE]
Exporting CA certificate signing request, please wait
The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-manage as:
ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate
The ipa-cacert-manage command was successful
```

- 如果外部 CA 不是 AD CS：

```
~]# ipa-cacert-manage renew --external-ca
Exporting CA certificate signing request, please wait
The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-manage as:
ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate
The ipa-cacert-manage command was successful
```

输出显示已创建了 CSR，并存储在 /var/lib/ipa/ca.csr 文件中。

2. 将位于 /var/lib/ipa/ca.csr 中的 CSR 提交到外部 CA。这个过程根据要用作外部 CA 的服务的不同而有所不同。

3. 检索发布的证书和 CA 证书链，用于在基本的 64 编码 Blob 中发布 CA，即：

- 如果外部 CA 不是 AD CS，则使用 PEM 文件。
- 如果外部 CA 是 AD CS，则为 Base_64 证书。

每个证书服务的进程都有所不同。通常，网页或通知电子邮件中的下载链接允许管理员下载所有需要的证书。

如果外部 CA 是 AD CS，且您已通过 Microsoft Windows 认证授权机构管理窗口使用已知模板提交 CSR，AD CS 会立即发出证书，且 Save Certificate 对话框会出现在 AD CS Web 界面中，询问如何保存颁发的证书。

4. 再次运行 ipa-cacert-manage renewal 命令，添加提供完整证书链所需的所有 CA 证书文

件。多次使用 --external-cert-file 选项，根据需要指定任意文件：

```
~]# ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-
cert-file=/path/to/external_ca_certificate_1 --external-cert-
file=/path/to/external_ca_certificate_2
```

5. 在所有 IdM 服务器和客户端中，使用来自服务器的证书更新本地 IdM 证书数据库：

```
[client ~]$ ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

6. （可选）检查您的更新是否成功，并将新 CA 证书添加到 /etc/ipa/ca.crt 文件中：

```
[client ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -print_certs
-text -noout
[...]
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 39 (0x27)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority
        Validity
            Not Before: Jul  1 16:32:45 2019 GMT
            Not After : Jul  1 16:32:45 2039 GMT
        Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority
[...]
```

输出显示更新已成功，因为新 CA 证书使用旧的 CA 证书列出。

# 第 72 章 IDM 离线时续订过期的系统证书

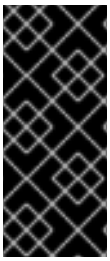
如果系统证书已过期，Identity Management(IdM)无法启动。IdM 支持使用 ipa-cert-fix 工具更新系统证书。

**先决条件**

- IdM 仅安装在 Red Hat Enterprise Linux 8.1 或更高的版本上。
- 通过在主机上输入 ipactl start --ignore-service-failures 命令来确保 LDAP 服务正在运行。

## 72.1. 在 CA 续订服务器上续订过期的系统证书

按照以下流程对过期的 IdM 证书应用 ipa-cert-fix 工具。



**重要**

如果您在不是 CA 续订服务器的 CA（证书授权机构）主机上运行 ipa-cert-fix 工具，并且 实用程序续订共享证书，则该主机会自动成为域中的新 CA 续订服务器。域中必须始终只有一个 CA 续订服务器，以避免不一致。

**先决条件**

- 使用管理权限登录到服务器

**流程**

1. （可选）备份系统。这强烈推荐，因为 ipa-cert-fix 对 nssdb 进行了不可逆的更改。因为 ipa-cert-fix 也对 LDAP 进行了更改，因此也建议备份整个集群。
2. 启动 ipa-cert-fix 工具，以分析系统并列出需要续订的过期证书：

```
# ipa-cert-fix
...
The following certificates will be renewed:

Dogtag sslserver certificate:
  Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
```

```
  Serial:  13
  Expires: 2019-05-12 05:55:47
...
Enter "yes" to proceed:
```

3. 输入 yes 以开始续订过程：

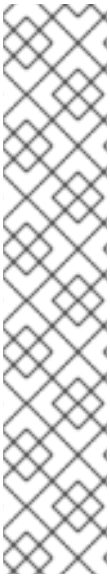
```
Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
  Serial:  268369925
  Expires: 2021-08-14 02:19:33
...

Becoming renewal master.
The ipa-cert-fix command was successful
```

ipa-cert-fix 更新所有过期证书前最多可能需要一分钟的时间。

4. （可选）验证所有服务现在是否都在运行：

```
# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

此时，证书已被续订，服务正在运行。下一步是检查 IdM 域中的其他服务器。



注意

如果您需要修复跨多个 CA 服务器的证书：

1. 确保 LDAP 复制在拓扑中正常工作后，根据上述流程，首先在一台 CA 服务器上运行 ipa-cert-fix。

2. 在另一台 CA 服务器上运行 ipa-cert-fix 之前，请通过 getcert-resubmit （在另一台 CA 服务器上）触发共享证书的 Certmonger 续订，以避免不必要的共享证书的续订。

## 72.2. 续订后验证 IDM 域中的其他 IDM 服务器

在使用 ipa-cert-fix 工具续订 CA 续订服务器证书后，您必须：

- 重新启动 域中的所有其他身份管理(IdM)服务器。
- 检查 certmonger 是否更新的证书。
- 如果有其他带有过期系统证书的证书颁发机构(CA)副本，还可以使用 ipa-cert-fix 工具续订这些证书。

**先决条件**

- 使用管理权限登录服务器。

**流程**

1. 使用 --force 参数重启 IdM：

```
# ipactl restart --force
```

使用 --force 参数时，ip actl 实用程序会忽略单个服务启动失败。例如，如果服务器也是证书过期的 CA，pki-tomcat 服务将无法启动。这是预期并忽略的，因为使用了 --force 参数。

2. 重启后，验证 certmonger 服务 是否已更新证书（certificate 状态显示 MONITORING）：

```
# getcert list | egrep '^Request|status:|subject:'
Request ID '20190522120745':
        status: MONITORING
        subject: CN=IPA RA,O=EXAMPLE.COM 201905222205
Request ID '20190522120834':
        status: MONITORING
        subject: CN=Certificate Authority,O=EXAMPLE.COM 201905222205
...
```

可能需要过些时间，certmonger 才会续订副本上的共享证书。

3. 如果服务器也是 CA，以上命令会报告 pki-tomcat 服务使用的证书的 CA_UNREACHABLE：

```
Request ID '20190522120835':
        status: CA_UNREACHABLE
        subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
...
```

4. 要续订此证书，请使用 ipa-cert-fix 工具：

```
# ipa-cert-fix
Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM
  Serial:  3
  Expires: 2019-05-11 12:07:11

Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
  Serial:  15
  Expires: 2019-08-14 04:25:05

The ipa-cert-fix command was successful
```

现在，所有 IdM 证书已被更新并可以正常工作。

# 第 73 章 如果 WEB 服务器和 LDAP 服务器证书还没有在 IDM 副本中过期，替换它们

作为身份管理(IdM)系统管理员，您可以手动替换运行在 IdM 服务器上的 web（或 httpd）和 LDAP（或 Directory）服务的证书。例如，如果证书接近到期，且 certmonger 程序没有配置为自动更新证书，或者证书由外部证书颁发机构(CA)签名，则可能需要这样做。

这个示例为运行在 server.idm.example.com IdM 服务器上的服务安装证书。您从外部 CA 获取证书。

**注意**

HTTP 和 LDAP 服务证书在不同的 IdM 服务器上有不同的密钥对和主题名称，因此您必须单独在每台 IdM 服务器上更新证书。

**先决条件**

- 在 IdM 服务器有复制协议的拓扑中至少有一个其他 IdM 副本，web 和 LDAP 证书仍然有效。这是 ipa-server-certinstall 命令的先决条件。该命令需要 TLS 连接与其他 IdM 副本通信。但是，使用无效证书时，这种连接无法建立，ipa-server-certinstall 命令将失败。在这种情况下，如果 整个 IdM 部署中已过期，请参阅替换 web 服务器和 LDAP 服务器证书。
- 您有访问 IdM 服务器的 root 权限。
- 您知道 目录管理器 密码。
- 您可以访问存储外部 CA 的 CA 证书链的文件 *ca_certificate_chain_file.crt* 。

**流程**

1. 将 *ca_certificate_chain_file.crt* 中包含的证书作为额外的 CA 证书安装到 IdM：

```
# ipa-cacert-manage install
```

2. 使用来自 *ca_certicate_chain_file.crt* 的证书更新本地 IdM 证书数据库：

```
# ipa-certupdate
```

3. 使用 OpenSSL 工具生成私钥和证书签名请求(CSR)：

```
$ openssl req -new -newkey rsa:4096 -days 365 -nodes -keyout new.key -out new.csr -addext "subjectAltName = DNS:server.idm.example.com" -subj '/CN=server.idm.example.com,O=IDM.EXAMPLE.COM'
```

将 CSR 提交给外部 CA。这个过程根据要用作外部 CA 的服务的不同而有所不同。在 CA 为证书签名后，将证书导入到 IdM 服务器。

4. 在 IdM 服务器上，将 Apache Web 服务器的旧私钥和证书替换为新密钥和新签名的证书：

```
# ipa-server-certinstall -w --pin=password new.key new.crt
```

在以上命令中：

- -w 选项指定您要将证书安装到 Web 服务器中。
- pin 选项指定保护私钥的密码。

5. 出现提示时，输入 目录管理器 密码。

6. 将 LDAP 服务器的旧私钥和证书替换为新密钥和新签名的证书：

```
# ipa-server-certinstall -d --pin=password new.key new.cert
```

在以上命令中：

- -d 选项指定您要将证书安装到 LDAP 服务器中。
- pin 选项指定保护私钥的密码。

7. 出现提示时，输入 目录管理器 密码。

8. 重启 httpd 服务：

```
# systemctl restart httpd.service
```

9. 重启 Directory 服务：

```
# systemctl restart dirsrv@IDM.EXAMPLE.COM.service
```

10. 如果在服务器上删除或替换了子 CA，请更新客户端：

```
# ipa-certupdate
```

**其它资源**

- 转换证书格式以便和 IdM 一起工作
- ipa-server-certinstall(1) 手册页

# 第 74 章 如果 WEB 服务器和 LDAP 服务器证书在整个 IDM 部署中已过期

身份管理(IdM)使用以下服务证书：

- LDAP （或 目录）服务器证书
- Web （或 httpd）服务器证书
- PKINIT 证书

在没有 CA 的 IdM 部署中，certmong er 默认不跟踪 IdM 服务证书或通知其过期。如果 IdM 系统管理员没有手动设置这些证书的通知，或者将 certmonger 配置为跟踪它们，则证书将在不通知的情况下过期。

按照以下流程，为运行在 server.idm.example.com IdM 服务器上的 httpd 和 LDAP 服务手动替换过期的证书。

**注意**

HTTP 和 LDAP 服务证书在不同 IdM 服务器上具有不同的密钥对和主题名称。因此，您必须单独更新每个 IdM 服务器中的证书。

**先决条件**

- 在拓扑中的 *所有* IdM 副本中，HTTP 和 LDAP 证书已过期。如果没有，请参阅 替换 web 服务器和 LDAP 服务器证书（如果它们还没有在 IdM 副本中过期 ）。
- 有到 IdM 服务器和副本的 root 访问权限。
- 您知道 目录管理器 密码。
- 您已创建以下目录和文件的备份：

- /etc/dirsrv/slapd-*IDM-EXAMPLE-COM*/
- /etc/httpd/alias
- /var/lib/certmonger
- /var/lib/ipa/certs/

**流程**

1. （可选）执行 /var/lib/ipa/private 和 /var/lib/ipa/passwds 的备份。
2. 如果您没有使用同一 CA 对新证书进行签名，或者已安装的 CA 证书不再有效，请使用包含外部 CA 的有效 CA 的文件更新本地数据库中外部 CA 的信息。文件在 PEM 和 DER 证书、PKCS#7 证书链、PKCS#8 和原始私钥和 PKCS#12 格式接受。

   a. 将 *ca_certificate_chain_file.crt* 中提供的证书作为额外的 CA 证书安装到 IdM 中：

   ```
   # ipa-cacert-manage install ca_certificate_chain_file.crt
   ```

   b. 使用来自 *ca_certicate_chain_file.crt* 的证书更新本地 IdM 证书数据库：

   ```
   # ipa-certupdate
   ```

3. 为 httpd 和 LDAP 请求证书：

   a. 使用 OpenSSL 工具，为在 IdM 实例上运行的 Apache Web 服务器创建到第三方 CA 的证书签名请求(CSR)。

   - 创建新私钥是可选的。如果您仍然有原始私钥，可以将 -in 选项与 openssl req 命令一起使用，以指定要从中读取请求的输入文件名：

   ```
   $ openssl req -new -nodes -in /var/lib/ipa/private/httpd.key -out /tmp/http.csr -
   addext 'subjectAltName = DNS:_server.idm.example.com_,
   ```

```
otherName:1.3.6.1.4.1.311.20.2.3;UTF8:HTTP/server.idm.example.com@IDM.EXAMPLE.COM' -subj '/O=IDM.EXAMPLE.COM/CN=server.idm.example.com'
```

- 如果要创建新密钥：

```
$ openssl req -new -newkey rsa:2048 -nodes -keyout /var/lib/ipa/private/httpd.key -out /tmp/http.csr -addext 'subjectAltName = DNS:server.idm.example.com, otherName:1.3.6.1.4.1.311.20.2.3;UTF8:HTTP/server.idm.example.com@IDM.EXAMPLE.COM' -subj '/O=IDM.EXAMPLE.COM/CN=server.idm.example.com'
```

b. 使用 OpenSSL 实用程序为 IdM 实例上运行的 LDAP 服务器创建证书签名请求(CSR)：

```
$ openssl req -new -newkey rsa:2048 -nodes -keyout ~/ldap.key -out /tmp/ldap.csr -addext 'subjectAltName = DNS:server.idm.example.com, otherName:1.3.6.1.4.1.311.20.2.3;UTF8:ldap/server.idm.example.com@IDM.EXAMPLE.COM' -subj '/O=IDM.EXAMPLE.COM/CN=server.idm.example.com'
```

c. 将 CSR、/tmp/http.csr 和 tmp/ldap.csr 提交到外部 CA，并获取 httpd 的证书和 LDAP 的证书。这个过程根据要用作外部 CA 的服务的不同而有所不同。

4. 为 httpd 安装证书：

```
# cp /path/to/httpd.crt /var/lib/ipa/certs/
```

5. 将 LDAP 证书安装到 NSS 数据库中：

a. [可选] 列出可用证书：

```
# certutil -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ -L
Certificate Nickname                                         Trust Attributes
                                                             SSL,S/MIME,JAR/XPI

Server-Cert                                                  u,u,u
```

默认证书 nickname 是 Server-Cert，但可能应用了不同的名称。

b. 使用上一步中的证书 nickname 从 NSS 数据库(NSSDB)中删除旧的无效证书：

```
# certutil -D -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ -n 'Server-Cert' -f
/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt
```

c. 创建 PKCS12 文件以简化导入过程到 NSSDB 中：

```
# openssl pkcs12 -export -in ldap.crt -inkey ldap.key -out ldap.p12 -name Server-
Cert
```

d. 将创建的 PKCS#12 文件安装到 NSSDB 中：

```
# pk12util -i ldap.p12 -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ -k
/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt
```

e. 检查新证书是否已成功导入：

```
# certutil -L -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/
```

6. 重启 httpd 服务：

```
# systemctl restart httpd.service
```

7. 重启 Directory 服务：

```
# systemctl restart dirsrv@IDM-EXAMPLE-COM.service
```

8. 在所有 IdM 副本中执行所有前面的步骤。这是在副本之间建立 TLS 连接的先决条件。

9. 将新证书注册到 LDAP 存储：

a. 将 Apache Web 服务器的旧私钥和证书替换为新密钥和新签名证书：

```
# ipa-server-certinstall -w --pin=password /var/lib/ipa/private/httpd.key
/var/lib/ipa/certs/httpd.crt
```

在以上命令中：

- -w 选项指定您要将证书安装到 Web 服务器中。
- pin 选项指定保护私钥的密码。

b. 出现提示时，输入 目录管理器 密码。

c. 将 LDAP 服务器的旧私钥和证书替换为新密钥和新签名的证书：

```
# ipa-server-certinstall -d --pin=password /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ldap.key /path/to/ldap.crt
```

在以上命令中：

- -d 选项指定您要将证书安装到 LDAP 服务器中。
- pin 选项指定保护私钥的密码。

d. 出现提示时，输入 目录管理器 密码。

e. 重启 httpd 服务：

```
# systemctl restart httpd.service
```

f. 重启 Directory 服务：

```
# systemctl restart dirsrv@IDM-EXAMPLE-COM.service
```

10. 在所有其他受影响的副本上执行上一步中的命令。

**其它资源**

将证书格式转换可与 IdM 一起工作 * man ipa-server-certinstall (1) *，如何在过期后，手动在 RHEL

## 8 上续订身份管理(IPA)证书？(CA-less IPA)

# 第 75 章 在 IDM CA 服务器中生成 CRL

如果您的 IdM 部署使用嵌入式证书颁发机构(CA)，您可能需要从一个 Identity Management(IdM)服务器中移动生成证书颁发机构列表(CRL)。例如，当您要将服务器迁移到另一个系统时，可能需要这样做。

仅配置一台服务器来生成 CRL。执行 CRL publisher 角色的 IdM 服务器通常与执行 CA 续订服务器角色的服务器相同，但这不是强制要求。在取消 CRL publisher 服务器前，选择并配置另一个服务器来执行 CRL publisher 服务器角色。

## 75.1. 在 IDM 服务器中停止 CRL 生成

要停止在 IdM CRL 发布程序服务器上生成证书撤销列表(CRL)，请使用 ipa-crlgen-manage 命令。在禁用生成前，请验证服务器是否确实生成 CRL。然后您可以禁用它。

### 先决条件

- 身份管理(IdM)服务器安装在 RHEL 8.1 系统或更新版本中。
- 您必须以 root 身份登录。

### 流程

1. 检查您的服务器是否正在生成 CRL:

```
[root@server ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

2. 停止在服务器上生成 CRL：

```
[root@server ~]# ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
```

```
CRL generation disabled on the local host. Please make sure to configure CRL
generation on another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

3. 检查服务器是否停止生成 CRL:

```
[root@server ~]# ipa-crlgen-manage status
```

服务器停止生成 CRL。下一步是在 IdM 副本上启用 CRL 生成。

## 75.2. 在 IDM 副本服务器中启动 CRL 生成

您可以使用 ipa-crlgen-manage 命令在 IdM CA 服务器上开始生成证书撤销列表(CRL)。

**先决条件**

- 身份管理(IdM)服务器安装在 RHEL 8.1 系统或更新版本中。
- RHEL 系统必须是 IdM 证书颁发机构服务器。
- 您必须以 root 身份登录。

**流程**

1. 开始生成 CRL：

```
[root@replica1 ~]# ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL
generation master.
The ipa-crlgen-manage command was successful
```

2. 检查是否生成 CRL：

```
[root@replica1 ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

### 75.3. 更改 CRL 更新间隔

默认情况下，证书撤销列表(CRL)文件由身份管理证书颁发机构(Idm CA)自动生成。您可以按照以下流程更改此间隔。

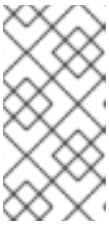
**流程**

1. 停止 CRL 生成服务器：

```
# systemctl stop pki-tomcatd@pki-tomcat.service
```

2. 打开 /var/lib/pki/pki-tomcat/conf/ca/CS.cfg 文件，并将 ca.crl.MasterCRL.autoUpdateInterval 值改为新的间隔设置。例如，每隔 60 分钟生成 CRL：

```
ca.crl.MasterCRL.autoUpdateInterval=60
```



**注意**

如果您更新了 ca.crl.MasterCRL.autoUpdateInterval 参数，则更改将在下一次计划的 CRL 更新后生效。

3. 启动 CRL 生成服务器：

```
# systemctl start pki-tomcatd@pki-tomcat.service
```

**其它资源**

- 有关 IdM 副本服务器上的 CRL 生成的更多信息，请参阅在 IdM 副本服务器上启动 CRL 生成。

# 第 76 章 停用执行 CA 续订服务器和 CRL 发布者角色的服务器

您可能有一台服务器同时执行证书颁发机构(CA)续订服务器角色和证书吊销列表(CRL)发布者角色。如果您需要将此服务器下线或停用，请选择并配置另一台 CA 服务器来执行这些角色。

在本例中，主机 server.idm.example.com ，其履行 CA 续订服务器和 CRL 发布者角色，必须停用。此流程将 CA 续订服务器和 CRL 发布者角色转移到主机 replica.idm.example.com，并从 IdM 环境中删除 server.idm.example.com。

**注意**

您不需要配置同一服务器来执行 CA 续订服务器和 CRL 发布者角色。

**先决条件**

- 有 IdM 管理员凭证。
- 您有要停用的服务器的 root 密码。
- 在您的 IdM 环境中至少有两个 CA 副本。

**流程**

1. 获取 IdM 管理员凭证：

```
[user@server ~]$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

2. *（可选）* 如果您不确定哪些服务器执行 CA 续订服务器和 CRL publisher 角色：

   a. 显示当前的 CA 续订服务器。您可以从任何 IdM 服务器运行以下命令：

```
[user@server ~]$ ipa config-show | grep 'CA renewal'
  IPA CA renewal master: server.idm.example.com
```

b.

测试主机是否为当前的 CRL 发布者。

```
[user@server ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

不生成 CRL 的 CA 服务器显示 CRL generation: disabled。

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: disabled
The ipa-crlgen-manage command was successful
```

继续在 CA 服务器上输入此命令，直到找到 CRL 发布者服务器。

c.

显示您可以提升的所有其他 CA 服务器，以履行这些角色。此环境有两个 CA 服务器。

```
[user@server ~]$ ipa server-role-find --role 'CA server'
----------------------
2 server roles matched
----------------------
  Server name: server.idm.example.com
  Role name: CA server
  Role status: enabled
  Server name: replica.idm.example.com
  Role name: CA server
  Role status: enabled
----------------------------
Number of entries returned 2
----------------------------
```

3.

将 replica.idm.example.com 设为 CA 续订服务器。

```
[user@server ~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com
```

4.

在 server.idm.example.com 上：

a.

禁用证书更新器任务：

```
[root@server ~]# pki-server ca-config-set ca.certStatusUpdateInterval 0
```

b. 重启 IdM 服务：

```
[root@server ~]# ipactl restart
```

5. 在 replica.idm.example.com 上：

a. 启用证书更新器任务：

```
[root@replica ~]# pki-server ca-config-unset ca.certStatusUpdateInterval
```

b. 重启 IdM 服务：

```
[root@replica ~]# ipactl restart
```

6. 在 server.idm.example.com 上，停止生成 CRL。

```
[user@server ~]$ ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL
generation on another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

7. 在 replica.idm.example.com 上，开始生成 CRL。

```
[user@replica ~]$ ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL
generation master.
The ipa-crlgen-manage command was successful
```

8. 停止 server.idm.example.com 上的 IdM 服务：

```
[root@server ~]# ipactl stop
```

9. 在 replica.idm.example.com 上，从 IdM 环境中删除 server.idm.example.com。

```
[user@replica ~]$ ipa server-del server.idm.example.com
```

10. 在 server.idm.example.com 上，以 root 帐户身份使用 ipa-server-install --uninstall 命令：

```
[root@server ~]# ipa-server-install --uninstall
...
Are you sure you want to continue with the uninstall procedure? [no]: yes
```

**验证步骤**

- 显示当前的 CA 续订服务器。

```
[user@replica ~]$ ipa config-show | grep 'CA renewal'
  IPA CA renewal master: replica.idm.example.com
```

- 确认 replica.idm.example.com 主机正在生成 CRL。

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

**其它资源**

- 更改和重置 IdM CA 续订服务器
- 在 IdM CA 服务器上生成 CRL
- 卸载 IdM 副本

# 第 77 章 使用 CERTMONGER 为服务获取 IDM 证书

## 77.1. CERTMONGER 概述

当 Identity Management(IdM)安装集成 IdM 证书颁发机构(CA)时，它将使用 certmonger 服务 来跟踪和续订系统和服务证书。当证书到达过期日期时，certmonger 通过以下 方法管理续订过程：

- 使用原始请求中提供的选项重新生成证书签名请求(CSR)。
- 使用 IdM API cert-request 命令将 CSR 提交到 IdM CA。
- 从 IdM CA 接收证书。
- 如果由原始请求指定，则执行预保存命令。
- 在续订请求中指定的位置安装新证书：在 NSS 数据库或文件中。
- 如果由原始请求指定，则执行保存后的命令。例如，保存后命令可以指示 certmonger 重新启动 相关服务，以便服务获取新证书。

证书类型 certmonger 跟踪

证书可分为系统和服务证书。

与服务证书（例如 HTTP、LDAP 和 PKINIT）不同，后者在不同服务器上具有不同的密钥对和主题名称，IdM 系统证书及其密钥由所有 CA 副本共享。IdM 系统证书包括：

- IdM CA 证书
- OCSP 签名证书
- IdM CA 子系统 证书

- IdM CA 审计签名 证书
- IdM 续订代理 (RA)证书
- KRA 传输和存储证书

certmonger 服务跟踪在安装带有集成 CA 的 IdM 环境期间请求的 IdM 系统和服务证书。Certmonger 还 跟踪系统管理员为 IdM 主机上运行的其他服务手动请求的证书。Certmonger 不会跟踪外部 CA 证书或用户证书。

### Certmonger 组件

certmonger 服务由两个主要组件组成：

- certmonger 守护进程，即引擎跟踪证书列表并启动续订命令
- 命令行界面 (CLI)的 getcert 实用程序允许系统管理员主动向 certmonger 守护进程 发送命令。

更具体来说，系统管理员可以使用 getcert 工具程序：

- 请求新证书
- 查看 certmonger 跟踪的证书 列表
- 启动或停止跟踪证书
- 续订证书

## 77.2. 使用 CERTMONGER 为服务获取 IDM 证书

为确保浏览器和身份管理(IdM)客户端上运行的 Web 服务之间的通信安全且加密，请使用 TLS 证书。从 IdM 证书颁发机构(CA)获取您的 Web 服务的 TLS 证书。

按照以下流程，使用 certmonger 获取在 IdM 客户端上运行的服务(HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM)的 IdM 证书。

使用 证书监控 器自动请求证书意味着，certmonger 在到期需要续订时管理和续订证书。

有关 certmonger 请求服务证书时所发生的情况的可视化表示，请参阅 请求服务证书的 certmonger 的通信流。

**先决条件**

- Web 服务器已注册为 IdM 客户端。
- 您有正在运行的 IdM 客户端的 root 访问权限。
- 请求证书的服务不必在 IdM 中预先存在。

**流程**

1. 在运行 HTTP 服务的 my_company.idm.example.com IdM 客户端中，请求与 HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM 主体对应的服务的证书，并指定：
   - 证书将存储在本地 /etc/pki/tls/certs/httpd.pem 文件中
   - 私钥存储在本地 /etc/pki/tls/private/httpd.key 文件中
   - 将 SubjectAltName 的 extensionRequest 添加到签名请求中，其 DNS 名称为 my_company.idm.example.com ：

```
# ipa-getcert request -K HTTP/my_company.idm.example.com -k
/etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
my_company.idm.example.com -C "systemctl restart httpd"
```

```
New signing request "20190604065735" added.
```

在以上命令中：

- ipa-getcert request 命令指定要从 IdM CA 获取证书。ipa-getcert request 命令是 getcert request -c IPA 的快捷方式。
- g 选项指定要生成的密钥的大小（如果尚未到位）。
- D 选项指定 要添加到请求的 SubjectAltName DNS 值。
- C 选项 指示 certmonger 在获取证书后重新启动 httpd 服务。
- 要指定证书与特定的配置集一起发布，请使用 -T 选项。
- 要使用指定的 CA 中的指定签发者请求证书，请使用 -X ISSUER 选项。

注意

RHEL 8 在 Apache 中使用与 RHEL 7 中使用的不同的 SSL 模块。SSL 模块依赖于 OpenSSL 而不是 NSS。因此，在 RHEL 8 中，您无法使用 NSS 数据库存储 HTTPS 证书和私钥。

2. （可选）检查请求的状态：

```
# ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
   status: MONITORING
   stuck: no
   key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
   certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
   CA: IPA
[...]
```

输出显示请求处于 MONITORING 状态，这表示已获取了证书。密钥对和证书的位置是请求的

位置。

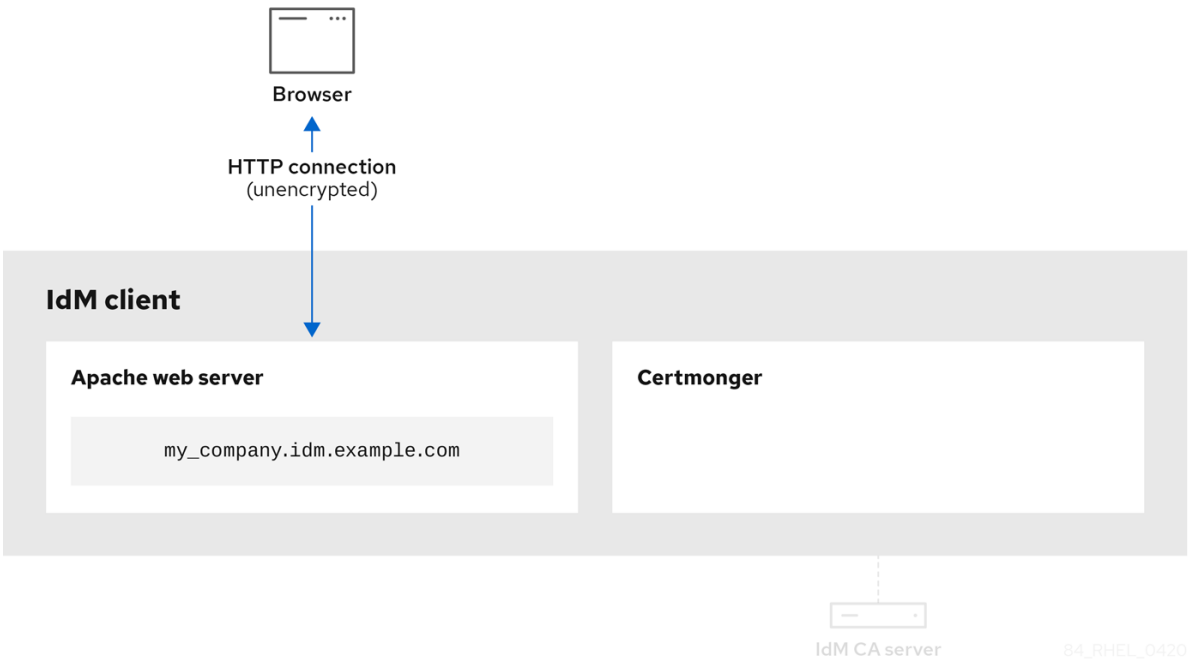
## 77.3. 请求服务证书的证书的通信流

这些图显示了当 certmonger 从身份管理(IdM)证书认证机构(CA)服务器请求服务证书时发生了什么情况的阶段。序列由这些图表组成：

- 未加密的通信
- 请求服务证书的 certmonger
- 发布服务证书的 IdM CA
- 应用服务证书的 certmonger
- 当旧的证书接近过期时，请求新证书的 certmonger

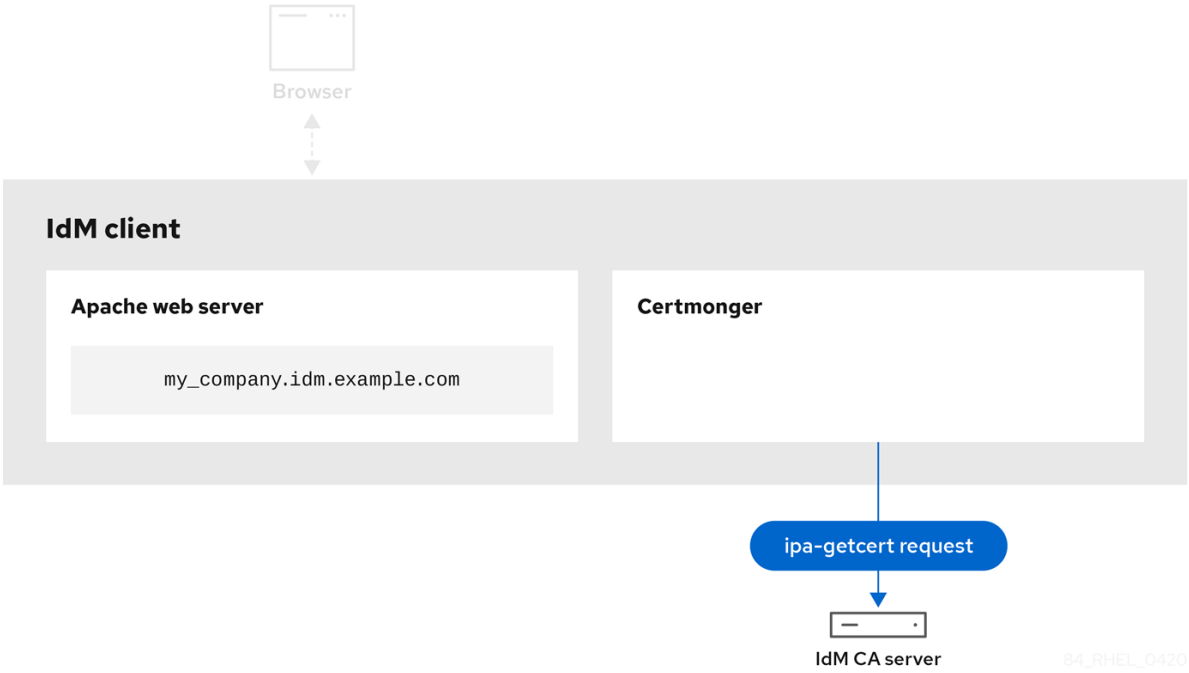
未加密的通信 显示初始情况：没有 HTTPS 证书，Web 服务器和浏览器之间的通信未加密。

图 77.1. 未加密的通信



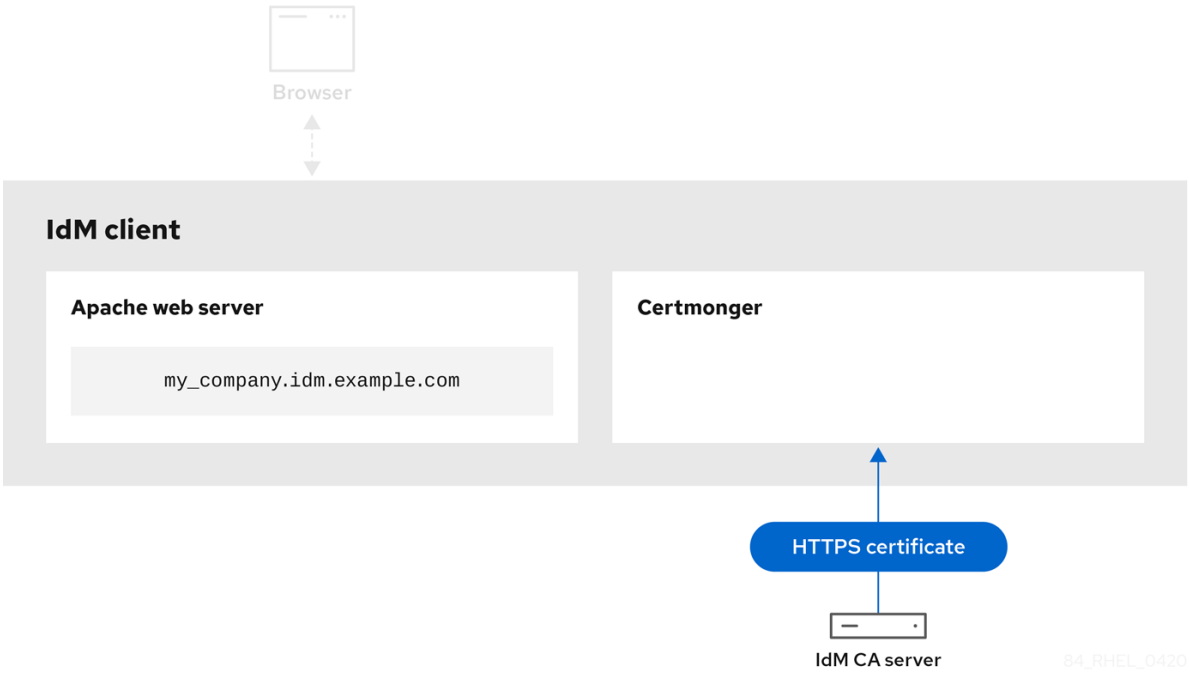
请求服务证书的 certmonger 显示系统管理员使用 certmonger 来手动为 Apache Web 服务器请求 HTTPS 证书。请注意，在请求 Web 服务器证书时，certmonger 不会直接与 CA 通信。它通过 IdM 代理。

图 77.2. 请求服务证书的 certmonger



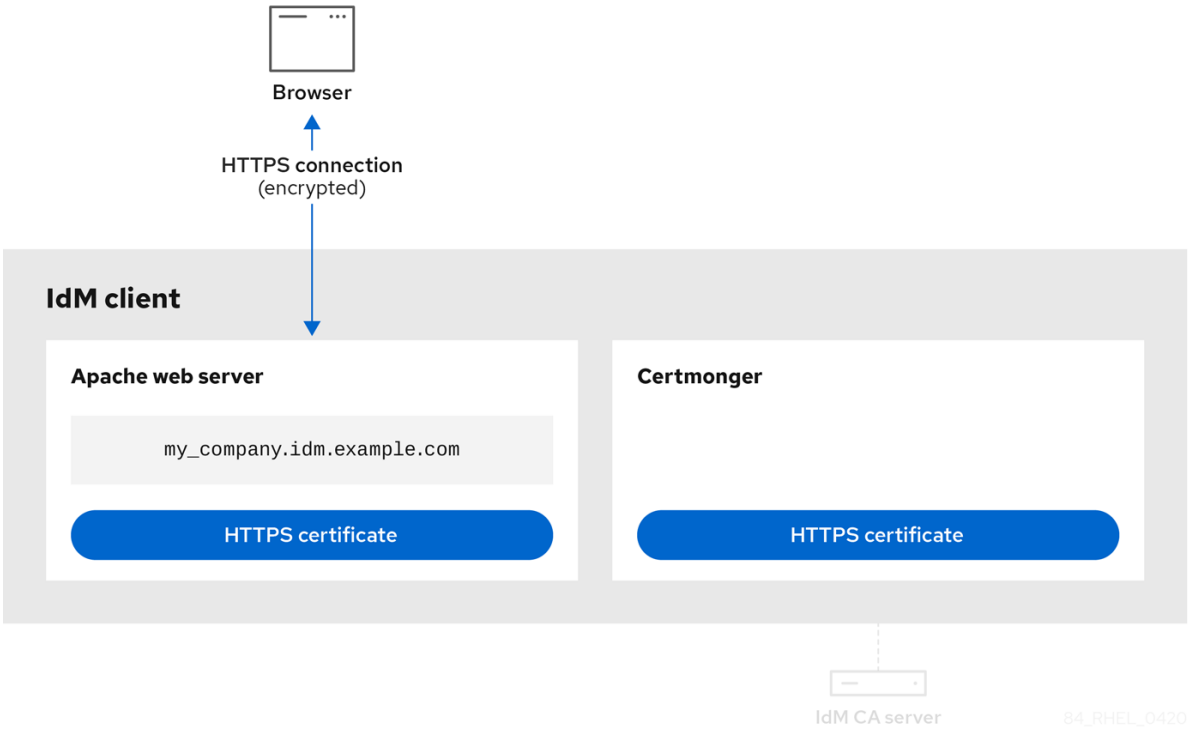

发布服务证书的IdM CA 显示为 web 服务器发布 HTTPS 证书的 IdM CA。

图 77.3. 发布服务证书的 IdM CA



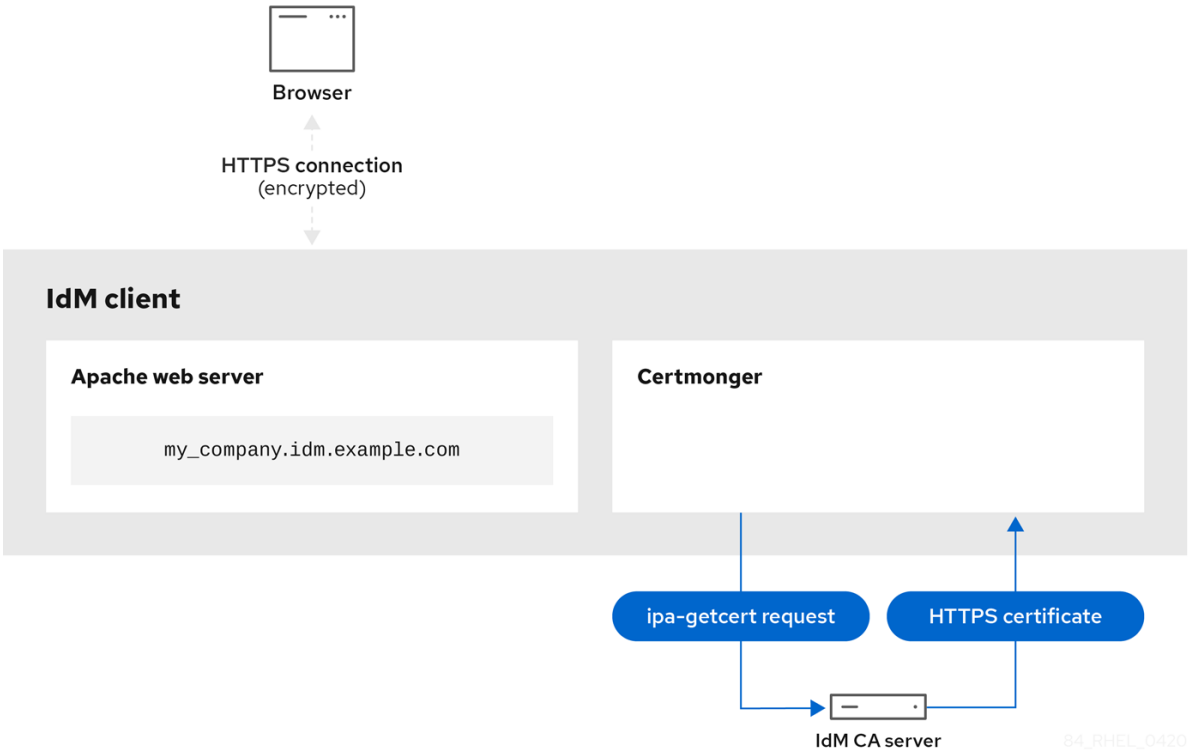

应用服务证书的 certmonger 显示将 HTTPS 证书放在 IdM 客户端上合适位置的 certmonger，如果指示要这样做，请重新启动 httpd 服务。随后 Apache 服务器使用 HTTPS 证书来加密自身和浏览器之间的流量。

图 77.4. 应用服务证书的 certmonger




当旧的证书接近过期时，请求新证书的 certmonger，显示 certmonger 在证书过期前自动从 IdM CA 请求续订服务证书。IdM CA 发布一个新证书。

图 77.5. 当旧的证书接近过期时，请求新证书的 certmonger




## 77.4. 查看由 CERTMONGER 跟踪的证书请求详情

certmonger 服务监控证书请求。成功签署证书请求后，会生成证书。Certmonger 管理证书请求，包括生成的证书。按照以下流程查看由 certmonger 管理的特定证书请求的详情。

**流程**

- 如果您知道如何指定证书请求，请只列出该特定证书请求的详细信息。例如，您可以指定：
  - 请求 ID
  - 证书的位置
  - 证书 nickname

  例如，要查看请求 ID 为 20190408143846 的证书详情，请使用 -v 选项查看您的证书请

求失败时的所有错误详情：

```
# getcert list -i 20190408143846 -v
Number of certificates and requests being tracked: 16.
Request ID '20190408143846':
 status: MONITORING
 stuck: no
 key pair storage: type=NSSDB,location='/etc/dirsrv/slapd-IDM-EXAMPLE-COM',nickname='Server-Cert',token='NSS Certificate DB',pinfile='/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt'
 certificate: type=NSSDB,location='/etc/dirsrv/slapd-IDM-EXAMPLE-COM',nickname='Server-Cert',token='NSS Certificate DB'
 CA: IPA
 issuer: CN=Certificate Authority,O=IDM.EXAMPLE.COM
 subject: CN=r8server.idm.example.com,O=IDM.EXAMPLE.COM
 expires: 2021-04-08 16:38:47 CEST
 dns: r8server.idm.example.com
 principal name: ldap/server.idm.example.com@IDM.EXAMPLE.COM
 key usage: digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment
 eku: id-kp-serverAuth,id-kp-clientAuth
 pre-save command:
 post-save command: /usr/libexec/ipa/certmonger/restart_dirsrv IDM-EXAMPLE-COM
 track: yes
 auto-renew: yes
```

输出显示有关证书的几段信息，例如：

- 证书位置；在上面的示例中，它是 /etc/dirsrv/slapd-IDM-EXAMPLE-COM 目录中的 NSS 数据库
- 证书 nickname；上例中为 Server-Cert
- 存储 pin 的文件；在上面的示例中，该文件为 /etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt
- 将用于续订证书的证书颁发机构(CA)；上例中为 IPA CA
- 到期日期；上例中为 2021-04-08 16:38:47 CEST
- 证书的状态；上例中，MON ITORING 状态表示证书有效并且被跟踪。

- 保存后的命令；在上面的示例中，这是 LDAP 服务的重新启动。

- 如果您不知道如何指定证书请求，请列出 certmonger 监控 或尝试获取的所有证书的详情：

```
# getcert list
```

其它资源

- 请参阅 getcert list 手册页。

## 77.5. 启动和停止证书跟踪

按照以下流程，使用 getcert stop-tracking 和 getcert start-tracking 命令来监控证书。这两个命令由 certmonger 服务提供。如果您已经从不同的 IdM 客户端上导入了身份管理(IdM)证书认证机构(CA)签发的证书，启用证书跟踪特别有用。启用证书跟踪也可以是以下置备方案的最后一步：

1. 在 IdM 服务器上，您可以为尚不存在的系统创建一个证书。
2. 您可以创建新系统。
3. 将新系统注册为 IdM 客户端。
4. 您可以将证书和密钥从 上的 IdM 服务器导入到 IdM 客户端。
5. 您开始使用 certmonger 来跟踪 证书，以确保其在过期时得到续订。

流程

- 使用 Request ID 20190408143846 禁用对证书的监控：

```
# getcert stop-tracking -i 20190408143846
```

有关更多选项，请参阅 getcert stop-tracking man page。

- 要启用对存储在 /tmp/some_cert.crt 文件中的证书的监控，其私钥存储在 /tmp/some_key.key 文件中：

```
# getcert start-tracking -c IPA -f /tmp/some_cert.crt -k /tmp/some_key.key
```

Certmonger 无法自动识别发布证书的 CA 类型。因此，如果 IdM CA 签发证书，将 -c 选项与 IPA 值一起添加到 getcert start-tracking 命令中。省略添加 -c 选项会导致 certmonger 进入 NEED_CA 状态。

有关更多选项，请参阅 getcert start-tracking man page。

**注意**

这两个命令不操作证书。例如，getcert stop-tracking 不会删除证书或将其从 NSS 数据库或文件系统中删除，只是将证书从受监控的证书列表中删除。同样，getcert start-tracking 只会在受监控证书列表中添加证书。

### 77.6. 手动续订证书

当证书即将到期时，certmonger 守护进程 会自动使用证书颁发机构(CA)帮助程序发出续订命令，获取更新的证书，并将上一个证书替换为新证书。

您还可以使用 getcert 重新提交 命令提前手动续订证书。这样，您可以更新证书包含的信息，例如，通过添加主题备用名称(SAN)。

按照以下流程手动续订证书。

**流程**

- 使用 Request ID 20190408143846 续订证书：

```
# getcert resubmit -i 20190408143846
```

要获取特定证书的 Request ID，请使用 getcert list 命令。详情请查看 getcert list man page。

## 77.7. 使CERTMONGER 恢复跟踪 CA 副本中的 IDM 证书

此流程演示了如何在跟踪 证书中断后使证书恢复 跟踪对使用集成证书颁发机构的 IdM 部署至关重要的 Identity Management(IdM)系统证书。中断可能是由在续订系统证书期间从 IdM 主机取消滚动，或者复制拓扑无法正常工作造成的。该程序还演示了如何使 证书恢复 跟踪 IdM 服务证书，即 HTTP、LDAP 和 PKINIT 证书。

**先决条件**

- 要恢复跟踪系统证书的主机是一个 IdM 服务器，它也是 IdM 证书颁发机构(CA)，而不是 IdM CA 续订服务器。

**流程**

1. 获取子系统 CA 证书的 PIN：

```
# grep 'internal=' /var/lib/pki/pki-tomcat/conf/password.conf
```

2. 添加跟踪到子系统 CA 证书，将以下命令中的 [internal PIN] 替换为上一步中获取的 PIN：

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "caSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"caSigningCert cert-pki-ca"' -T caCACert

# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "auditSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"auditSigningCert cert-pki-ca"' -T caSignedLogCert

# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "ocspSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"ocspSigningCert cert-pki-ca"' -T caOCSPCert

# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "subsystemCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"subsystemCert cert-pki-ca"' -T caSubsystemCert

# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "Server-Cert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"Server-Cert cert-pki-ca"' -T caServerCert
```

3. 添加对剩余的 IdM 证书、HTTP、LDAP 、 IPA 续订代理 和 PKINIT 证书的 跟踪：

```
# getcert start-tracking -f /var/lib/ipa/certs/httpd.crt -k /var/lib/ipa/private/httpd.key -p
/var/lib/ipa/passwds/idm.example.com-443-RSA -c IPA -C
/usr/libexec/ipa/certmonger/restart_httpd -T calPAserviceCert

# getcert start-tracking -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM -n "Server-Cert" -c IPA
-p /etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt -C
'/usr/libexec/ipa/certmonger/restart_dirsrv "IDM-EXAMPLE-COM"' -T calPAserviceCert

# getcert start-tracking -f /var/lib/ipa/ra-agent.pem -k /var/lib/ipa/ra-agent.key -c
dogtag-ipa-ca-renew-agent -B /usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_ra_cert -T caSubsystemCert

# getcert start-tracking -f /var/kerberos/krb5kdc/kdc.crt -k
/var/kerberos/krb5kdc/kdc.key -c dogtag-ipa-ca-renew-agent -B
/usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_kdc_cert -T KDCs_PKINIT_Certs
```

4. 重启证书 ：

```
# systemctl restart certmonger
```

5. 等待一分钟，直到 certmonger 启动后，然后检查新证书的状态：

```
# getcert list
```

**其它资源**

- 如果您的 IdM 系统证书已全部过期，请参阅 这个以知识为中心的支持(KCS)解决方案，来手动更新 IdM CA 服务器上的 IdM 系统证书，该服务器也是 CA 续订服务器和 CRL 发布者服务器。然后按照 这个 KCS 解决方案 中的步骤在拓扑中的所有其他 CA 服务器中手动续订 IdM 系统证书。

### 77.8. 使用 SCEP 和 CERTMONGER

简单证书注册协议(SCEP)是可在不同设备和操作系统中使用的证书管理协议。如果您在环境中使用 SCEP 服务器作为外部证书颁发机构(CA)，您可以使用 certmonger 获取 Identity Management(IdM)客户端的证书。

#### 77.8.1. SCEP 概述

简单证书注册协议(SCEP)是可在不同设备和操作系统中使用的证书管理协议。您可以使用 SCEP 服务

器作为外部证书颁发机构(CA)。

您可以配置 Identity Management(IdM)客户端，以直接从 CA SCEP 服务通过 HTTP 请求并检索证书。此过程由共享 secret 保护，该 secret 通常仅对有限时间有效。

在客户端，SCEP 要求您提供以下组件：

- SCEP URL：CA SCEP 接口的 URL。
- SCEP 共享 secret：在 CA 和 SCEP 客户端之间共享 质询密码 PIN，用于获取证书。

然后，客户端通过 SCEP 检索 CA 证书链，并将证书签名请求发送到 CA。

使用 certmonger 配置 SCEP 时，您可以创建一个新的 CA 配置配置文件，该配置文件指定了签发的证书参数。

### 77.8.2. 通过 SCEP 请求 IdM CA 签名证书

以下示例将 SCEP_example SCEP CA 配置添加到 certmonger，并在 client.idm.example.com IdM 客户端上请求新证书。certmonger 支持 NSS 证书数据库格式和基于文件的(PEM)格式，如 OpenSSL。

**先决条件**

- 您知道 SCEP URL。
- 您有 challengePassword PIN 共享 secret。

**流程**

1. 将 CA 配置添加到 certmonger ：

```
[root@client.idm.example.com ~]# getcert add-scep-ca -c SCEP_example -u
SCEP_URL
```

- -c: CA 配置强制别名。稍后可以将相同的值用于其他 getcert 命令。

- -u:服务器的 SCEP 接口的 URL。

重要

使用 HTTPS URL 时，还必须使用 -R 选项指定 SCEP 服务器 CA 证书的 PEM 格式副本的位置。

2. 验证 CA 配置是否已成功添加：

```
[root@client.idm.example.com ~]# getcert list-cas -c SCEP_example
CA 'SCEP_example':
    is-default: no
    ca-type: EXTERNAL
    helper-location: /usr/libexec/certmonger/scep-submit -u
http://SCEP_server_enrollment_interface_URL
    SCEP CA certificate thumbprint (MD5): A67C2D4B 771AC186 FCCA654A
5E55AAF7
    SCEP CA certificate thumbprint (SHA1): FBFF096C 6455E8E9 BD55F4A5 5787C43F
1F512279
```

如果成功添加了配置，certmonger 从远程 CA 检索 CA 链。然后，CA 链在命令输出中显示为 thumbprints。当通过未加密的 HTTP 访问服务器时，手动将 thumbprints 与 SCEP 服务器中显示的 thumbprints 进行比较，以防止 man-in-the-middle 攻击。

3. 从 CA 请求证书：

- 如果您使用 NSS：

```
[root@client.idm.example.com ~]# getcert request -I Example_Task -c
SCEP_example -d /etc/pki/nssdb -n ExampleCert -N cn="client.idm.example.com" -
L one-time_PIN -D client.idm.example.com
```

您可以使用选项来指定证书请求的以下参数：

  - 任务的 -I:（可选） 名称：请求的跟踪 ID。稍后可以将相同的值用于 getcert list 命令。

- -c ：将请求提交到的 CA 配置。
- -d ：包含 NSS 数据库的目录来存储证书和密钥。
- -n ：证书 Nickname，在 NSS 数据库中使用。
- -n: CSR 中的 Subject 名称。
- -L ： CA 发布时限一次性 质询Password PIN。
- - d：证书的主题备用名称，通常与主机名相同。

- 如果您使用 OpenSSL：

```
[root@client.idm.example.com ~]# getcert request -I Example_Task -c
SCEP_example -f /etc/pki/tls/certs/server.crt -k /etc/pki/tls/private/private.key -N
cn="client.idm.example.com" -L one-time_PIN -D client.idm.example.com
```

您可以使用选项来指定证书请求的以下参数：

- 任务的 -I:（可选） 名称：请求的跟踪 ID。稍后可以将相同的值用于 getcert list 命令。
- -c ：将请求提交到的 CA 配置。
- -f ：到证书的存储路径。
- -k:到密钥的存储路径。

- **-n: CSR 中的 Subject 名称。**
- **-L ： CA 发布时限一次性 质询Password PIN。**
- **- d：证书的主题备用名称，通常与主机名相同。**

验证

1. 验证证书**是否已颁发并正确存储在本地数据库中：**

- **如果您使用了 NSS，请输入：**

```
[root@client.idm.example.com ~]# getcert list -I Example_Task
 Request ID 'Example_Task':
        status: MONITORING
        stuck: no
        key pair storage:
type=NSSDB,location='/etc/pki/nssdb',nickname='ExampleCert',token='NSS Certificate
DB'
        certificate:
type=NSSDB,location='/etc/pki/nssdb',nickname='ExampleCert',token='NSS Certificate
DB'
        signing request thumbprint (MD5): 503A8EDD DE2BE17E 5BAA3A57 D68C9C1B
        signing request thumbprint (SHA1): B411ECE4 D45B883A 75A6F14D 7E3037F1
D53625F4
        CA: IPA
        issuer: CN=Certificate Authority,O=EXAMPLE.COM
        subject: CN=client.idm.example.com,O=EXAMPLE.COM
        expires: 2018-05-06 10:28:06 UTC
        key usage: digitalSignature,keyEncipherment
        eku: iso.org.dod.internet.security.mechanisms.8.2.2
        certificate template/profile: IPSECIntermediateOffline
        pre-save command:
        post-save command:
        track: yes
 auto-renew: yes
```

- **如果使用 OpenSSL，请输入：**

```
[root@client.idm.example.com ~]# getcert list -I Example_Task
Request ID 'Example_Task':
        status: MONITORING
        stuck: no
        key pair storage: type=FILE,location='/etc/pki/tls/private/private.key'
```

```
certificate: type=FILE,location='/etc/pki/tls/certs/server.crt'
CA: IPA
issuer: CN=Certificate Authority,O=EXAMPLE.COM
subject: CN=client.idm.example.com,O=EXAMPLE.COM
expires: 2018-05-06 10:28:06 UTC
eku: id-kp-serverAuth,id-kp-clientAuth
pre-save command:
post-save command:
track: yes
auto-renew: yes
```

状态 MONITORING 表示成功检索签发的证书。getcert-list(1) man page 列出了其他可能的状态及其含义。

**其它资源**

- 有关请求证书时的更多信息，请参阅 getcert-request(1) man page。

### 77.8.3. 使用 certmonger 自动续订 AD SCEP 证书

当 certmonger 发送 SCEP 证书签名请求时，此请求将使用现有证书私钥签名。但是，默认由 certmonger 发送的续订请求还包括用于获取证书的 challengePassword PIN。

作为 SCEP 服务器的 Active Directory(AD)网络设备注册服务(NDES)服务器会自动拒绝包含原始 challengePassword PIN 的续订请求。因此，续订会失败。

要与 AD 续订，您需要配置 certmonger，以在没有 challengePassword PIN 的情况下发送签名的续订请求。您还需要配置 AD 服务器，使其不会在续订时比较主题名称。

**注意**

AD 以外的 SCEP 服务器也会拒绝包含 challengePassword 的请求。在这些情况下，您可能还需要以这种方式更改 certmonger 配置。

**先决条件**

- RHEL 服务器必须正在运行 RHEL 8.6 或更新版本。

**流程**

1. 在 AD 服务器上打开 regedit。

2. 在 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\MSCEP 子键中，添加一个新的 32 位 REG_DWORD 条目 DisableRenewalSubjectNameMatch，并将其值设为 1。

3. 在运行 certmonger 的服务器上，打开 /etc/certmonger/certmonger.conf 文件，并添加以下部分：

```
[scep]
challenge_password_otp = yes
```

4. 重启 certmonger：

```
# systemctl restart certmonger
```

# 第 78 章 使用 RHEL 系统角色请求证书

您可以使用 证书系统角色 发布和管理证书。

## 78.1. CERTIFICATE RHEL 系统角色

使用 certificate 系统角色，您可以使用 Ansible Core 管理发布和更新 TLS 和 SSL 证书。

该角色使用 certmonger 作为 证书提供程序，目前支持发布和续订自签名证书并使用 IdM 集成认证机构(CA)。

您可以将 Ansible playbook 中的以下变量与 证书系统角色 搭配使用：

certificate_wait

来指定任务是否应该等待要发布的证书。

certificate_requests

来表示要发布的每个证书及其参数。

其它资源

- /usr/share/ansible/roles/rhel-system-roles.certificate/README.md 文件
- /usr/share/doc/rhel-system-roles/certificate/ directory

## 78.2. 使用 CERTIFICATE RHEL 系统角色请求新的自签名证书

使用 certificate 系统角色，您可以使用 Ansible Core 发布自签名证书。

此过程使用 certmonger 供应商，并通过 getcert 命令请求证书。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 用于连接到受管节点的帐户具有 sudo 权限。

**流程**

1. 创建包含以下内容的 playbook 文件，如 ~/playbook.yml ：

```
---
- hosts: managed-node-01.example.com
  roles:
    - rhel-system-roles.certificate
  vars:
    certificate_requests:
      - name: mycert
        dns: "*.example.com"
        ca: self-sign
```

   - 将 name 参数设置为证书的所需名称，如 mycert。
   - 将 dns 参数设置为证书中包含的域，如 *.example.com。
   - 将 ca 参数设置为 self-sign。

   默认情况下，certmonger 会在证书过期前自动尝试续订证书。您可以通过将 Ansible playbook 中的 auto_renew 参数设置为 no 来禁用此功能。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

   请注意，这个命令只验证语法，不会防止错误但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

**其它资源**

- /usr/share/ansible/roles/rhel-system-roles.certificate/README.md 文件
- /usr/share/doc/rhel-system-roles/certificate/ directory

## 78.3. 使用 CERTIFICATE RHEL 系统角色从 IDM CA 请求一个新证书

使用证书系统角色，您可以在使用带有集成证书颁发机构(CA)的 IdM 服务器时，使用 anible-core 来发布证书。因此，当使用 IdM 作为 CA 时，您可以高效且一致地管理多个系统的证书信任链。

此过程使用 certmonger 供应商，并通过 getcert 命令请求证书。

**先决条件**

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 用于连接到受管节点的帐户具有 sudo 权限。

**流程**

1. 创建包含以下内容的 playbook 文件，如 ~/playbook.yml ：

```
---
- hosts: managed-node-01.example.com
  roles:
    - rhel-system-roles.certificate
  vars:
    certificate_requests:
      - name: mycert
```

```
    dns: www.example.com
    principal: HTTP/www.example.com@EXAMPLE.COM
    ca: ipa
```

- 将 name 参数设置为证书的所需名称，如 mycert。
- 将 dns 参数设置为证书中包含的域，如 www.example.com。
- 将 principal 参数设置为指定 Kerberos 主体，如 HTTP/www.example.com@EXAMPLE.COM。
- 将 ca 参数设置为 ipa。

默认情况下，certmonger 会在证书过期前自动尝试续订证书。您可以通过将 Ansible playbook 中的 auto_renew 参数设置为 no 来禁用此功能。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不会防止错误但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

**其它资源**

- /usr/share/ansible/roles/rhel-system-roles.certificate/README.md 文件
- /usr/share/doc/rhel-system-roles/certificate/ directory

## 78.4. 使用证书 RHEL 系统角色指定在证书颁发前或之后要运行的命令

使用 证书 角色，您可以使用 Ansible Core 在签发或更新证书前和之后执行命令。

在以下示例中，管理员确保在为 www.example.com 发布或更新自签名证书前停止 httpd 服务，然后再重启该服务。

**先决条件**

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 用于连接到受管节点的帐户具有 sudo 权限。

**流程**

1. 创建包含以下内容的 playbook 文件，如 ~/playbook.yml ：

```
---
- hosts: managed-node-01.example.com
  roles:
    - rhel-system-roles.certificate
  vars:
    certificate_requests:
      - name: mycert
        dns: www.example.com
        ca: self-sign
        run_before: systemctl stop httpd.service
        run_after: systemctl start httpd.service
```

- 将 name 参数设置为证书的所需名称，如 mycert。
- 将 dns 参数设置为证书中包含的域，如 www.example.com。
- 将 ca 参数设置为您要用来发布证书的 CA，如 自签名。
- 将 run_before 参数设置为在签发或续订证书之前要执行的命令，如 systemctl stop httpd.service。

- 将 run_after 参数设置为在签发或续订此证书后要执行的命令，如 systemctl start httpd.service。

默认情况下，certmonger 会在证书过期前自动尝试续订证书。您可以通过将 Ansible playbook 中的 auto_renew 参数设置为 no 来禁用此功能。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不会防止错误但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

**其它资源**

- /usr/share/ansible/roles/rhel-system-roles.certificate/README.md 文件
- /usr/share/doc/rhel-system-roles/certificate/ directory

# 第 79 章 将应用程序限制为只信任证书子集

如果您的 Identity Management(IdM)安装配置了集成证书系统(CS)证书颁发机构(CA)，您可以创建轻量级子 CA。您创建的所有子 CA 都从属于证书系统的主 CA，即 ipa CA。

*在这种情况下，轻量 级子 CA 意味着为特定目的发布证书的子 CA*。例如，轻量级子 CA 允许您配置服务，如虚拟专用网络(VPN)网关和 Web 浏览器，以仅接受 *子 CA A* 发布的证书。通过将其他服务配置为仅接受由 *子 CA B* 发布的证书，您可以防止它们接受 *子 CA*、*主 CA*（即 ipa CA）和两者之间的任何中间子 CA 发布的证书。

如果您撤销了子 CA 的中间证书，则正确配置的客户端会自动将此子 CA 发布的所有证书视为无效。所有其他直接由 root CA、ipa 或其他 子 CA 发布的证书保持有效。

本节使用 Apache Web 服务器的示例来说明如何将应用限制为仅信任某一证书子集。完成本节以限制 IdM 客户端上运行的 Web 服务器使用 webserver-ca IdM 子 CA 发布的证书，并要求用户使用 web client-ca IdM 子 CA 发布的用户证书向 web 服务器进行身份验证。

您需要执行的步骤有：

1. 创建 IdM 子 CA
2. 从 IdM WebUI 下载子 CA 证书
3. 创建 CA ACL，指定正确组合用户、服务和 CA，以及使用的证书配置集
4. 从 IdM 子 CA 请求在 IdM 客户端上运行的 web 服务的证书
5. 设置单实例 Apache HTTP 服务器
6. 向 Apache HTTP 服务器添加 TLS 加密
7. 在 Apache HTTP 服务器中设置支持的 TLS 协议版本

8. 在 Apache HTTP 服务器上设置受支持的密码

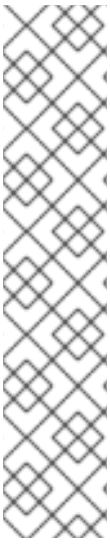
9. 在 web 服务器中配置 TLS 客户端证书身份验证

10. 从 IdM 子 CA 请求用户的证书，并将其导出到客户端

11. 将用户证书导入到浏览器中，并将浏览器配置为信任子 CA 证书

## 79.1. 管理轻量级子 CA

本节描述了如何管理轻量级从属证书颁发机构(sub-CA)。您创建的所有子 CA 都从属到证书系统的主 CA ， ipa CA。您还可以禁用和删除子 CA。



**注意**

- 如果您删除了子 CA，则该子 CA 的吊销检查将不再工作。只有当子 CA 不再发布证书，且其 notAfter 过期时间在未来时，才可以删除该子 CA。
- 只有当子 CA 发布的证书仍然未过期时，才应禁用该子CA。如果子 CA 发布的所有证书都已过期，您可以删除该子 CA。
- 您不能禁用或删除 IdM CA。

有关管理子 CA 的详情，请参阅：

- 从 IdM WebUI 创建子 CA
- 从 IdM WebUI 删除子 CA
- 从 IdM CLI 创建子 CA

- 从 IdM CLI 禁用子 CA
- 从 IdM CLI 删除子 CA

### 79.1.1. 从 IdM WebUI 创建子 CA

按照以下流程，使用 IdM WebUI 创建名为 webserver-ca 和 webclient-ca 的新子 CA。

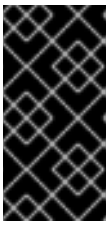
**先决条件**

- 确保您已获取管理员的凭据。

**流程**

1. 在 身份验证 菜单中，单击 Certificates。
2. 选择 证书授权 并单击 添加。
3. 输入 webserver-ca 子 CA 的名称。在 Subject DN 字段中输入 Subject DN，如 CN=WEBSERVER,O=IDM.EXAMPLE.COM。请注意，主题 DN 在 IdM CA 基础架构中必须是唯一的。
4. 输入 webclient-ca 子 CA 的名称。在 Subject DN 字段中输入 Subject DN CN=WEBCLIENT,O=IDM.EXAMPLE.COM。
5. 在命令行界面中，运行 ipa-certupdate 命令，来为 webserver-ca 和 webclient-ca 子 CA 证书创建 certmonger 追踪请求：

```
[root@ipaserver ~]# ipa-certupdate
```



重要

在创建子 CA 后忘记运行 ipa-certupdate 命令意味着，如果子 CA 证书过期，则子 CA 发布的最终用户证书将被视为无效，即使最终用户证书还没有过期。

验证

- 验证新子 CA 的签名证书是否已添加到 IdM 数据库中：

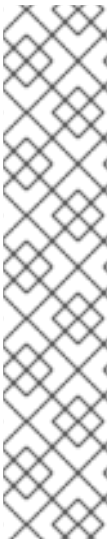
```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L

Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

caSigningCert cert-pki-ca               CTu,Cu,Cu
Server-Cert cert-pki-ca                 u,u,u
auditSigningCert cert-pki-ca            u,u,Pu
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc u,u,u
ocspSigningCert cert-pki-ca             u,u,u
subsystemCert cert-pki-ca               u,u,u
```

注意

新子 CA 证书自动传输到安装有证书系统实例的所有副本。

### 79.1.2. 从 IdM WebUI 删除子 CA

按照以下流程删除 IdM Web UI 中的轻量级子 CA。



注意

- 如果您删除了子 CA，则该子 CA 的吊销检查将不再工作。只有当子 CA 不再发布证书，且其 notAfter 过期时间在未来时，才可以删除该子 CA。
- 只有当子 CA 发布的证书仍然未过期时，才应禁用该子CA。如果子 CA 发布的所有证书都已过期，您可以删除该子 CA。
- 您不能禁用或删除 IdM CA。

先决条件

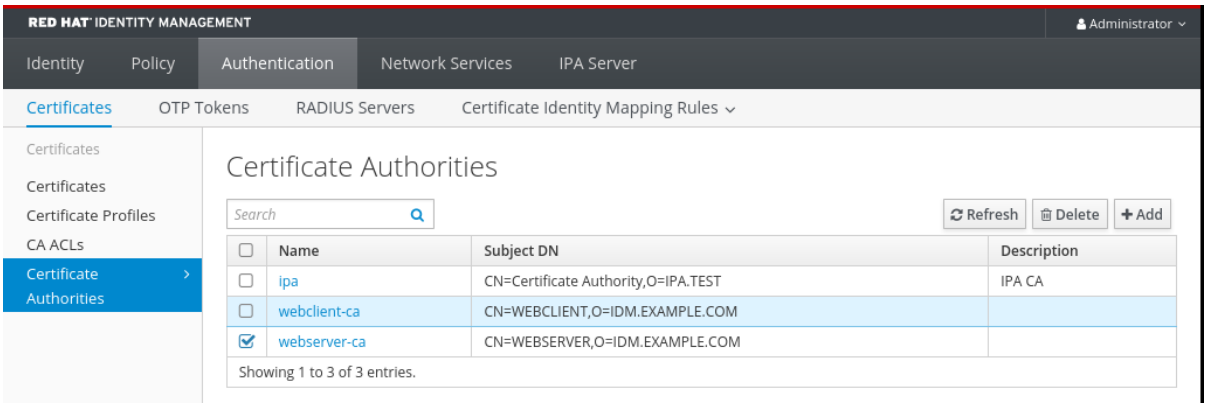
- 确保您已获取管理员的凭据。

- 您已在 IdM CLI 中禁用了子 CA。请参阅 从 IdM CLI 禁用子 CA

**流程**

1. 在 IdM Web UI 中，打开 身份验证 选项卡，然后选择 证书 子选项卡。
2. 选择 证书颁发机构 。
3. 选择要删除的子 CA，然后单击" 删除 "。

图 79.1. 在 IdM Web UI 中删除子 CA




4. 单击 删除 以确认。

子 CA 从 证书颁发机构 列表中删除。

### 79.1.3. 从 IdM CLI 创建子 CA

按照以下流程，使用 IdM CLI 创建名为 webserver-ca 和 webclient-ca 的新子 CA。

**先决条件**

- 确保您已获取管理员的凭据。
- 确保您已登录到 CA 服务器的 IdM 服务器。

**流程**

1. 输入 **ipa ca-add** 命令，再指定 **webserver-ca** 子 **CA** 的名称及其 **Subject Distinguished Name(DN)**：

```
[root@ipaserver ~]# ipa ca-add webserver-ca --
subject="CN=WEBSERVER,O=IDM.EXAMPLE.COM"
------------------
Created CA "webserver-ca"
------------------
  Name: webserver-ca
  Authority ID: ba83f324-5e50-4114-b109-acca05d6f1dc
  Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

**Name**

CA 的名称。

授权 ID

自动创建 CA 独立 ID。

主题 DN

主题区分名称(DN)。主题 DN 在 IdM CA 基础架构中必须是唯一的。

签发者 DN

签发子 CA 证书的父 CA。所有子 CA 都是作为 IdM root CA 的子 CA 创建的。

2. 创建 **webclient-ca** 子 **CA** 以向 **Web** 客户端发布证书：

```
[root@ipaserver ~]# ipa ca-add webclient-ca --
subject="CN=WEBCLIENT,O=IDM.EXAMPLE.COM"
------------------
Created CA "webclient-ca"
------------------
  Name: webclient-ca
  Authority ID: 8a479f3a-0454-4a4d-8ade-fd3b5a54ab2e
  Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

3. 运行 **ipa-certupdate** 命令，来为 **webserver-ca** 和 **webclient-ca** 子 **CAs** 证书创建 **certmonger** 追踪请求：

```
[root@ipaserver ~]# ipa-certupdate
```

重要

如果您在创建子 CA 后忘记了运行 ipa-certupdate 命令，且子 CA 证书已过期，则该子 CA 发布的最终身份证书被视为无效，即使最终身份证书没有过期。

验证步骤

- 验证新子 CA 的签名证书是否已添加到 IdM 数据库中：

```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L

Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

caSigningCert cert-pki-ca               CTu,Cu,Cu
Server-Cert cert-pki-ca                 u,u,u
auditSigningCert cert-pki-ca            u,u,Pu
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc u,u,u
ocspSigningCert cert-pki-ca             u,u,u
subsystemCert cert-pki-ca               u,u,u
```

注意

新子 CA 证书自动传输到安装有证书系统实例的所有副本。

### 79.1.4. 从 IdM CLI 禁用子 CA

按照以下流程，从 IdM CLI 禁用子 CA。如果子 CA 发布的证书还有未过期的，则您不应该删除它，但可以禁用它。如果您删除了子 CA，则对该子 CA 的吊销检查将不再工作。

先决条件

- 确保您已获取管理员的凭据。

流程

1. 运行 ipa ca-find 命令来确定您要删除的子 CA 的名称：

```
[root@ipaserver ~]# ipa ca-find
```

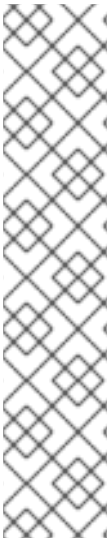
```
-------------
3 CAs matched
-------------
  Name: ipa
  Description: IPA CA
  Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
  Subject DN: CN=Certificate Authority,O=IPA.TEST
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

  Name: webclient-ca
  Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
  Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

 Name: webserver-ca
  Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
  Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST
----------------------------
Number of entries returned 3
----------------------------
```

2. 运行 **ipa ca-disable** 命令来禁用您的子 CA，在本例中为 **webserver-ca** ：

```
ipa ca-disable webserver-ca
-------------------------
Disabled CA "webserver-ca"
-------------------------
```

### 79.1.5. 从 IdM CLI 删除子 CA

按照以下流程从 IdM CLI 删除轻量级子 CA。



**注意**

- 如果您删除了子 CA，则该子 CA 的吊销检查将不再工作。只有当子 CA 不再发布证书，且其 notAfter 过期时间在未来时，才可以删除该子 CA。
- 只有当子 CA 发布的证书仍然未过期时，才应禁用该子CA。如果子 CA 发布的所有证书都已过期，您可以删除该子 CA。
- 您不能禁用或删除 IdM CA。

**先决条件**

- 确保您已获取管理员的凭据。

**流程**

1. 要显示子 CA 和 CA 的列表，请运行 ipa ca-find 命令：

```
# ipa ca-find
----------
3 CAs matched
----------
  Name: ipa
  Description: IPA CA
  Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
  Subject DN: CN=Certificate Authority,O=IPA.TEST
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

  Name: webclient-ca
  Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
  Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

 Name: webserver-ca
  Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
  Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST
----------------------------
Number of entries returned 3
----------------------------
```

2. 运行 ipa ca-disable 命令来禁用您的子 CA，在本例中为 webserver-ca ：

```
# ipa ca-disable webserver-ca
-------------------------
Disabled CA "webserver-ca"
-------------------------
```

3. 删除子 CA，在本例中为 webserver-ca ：

```
# ipa ca-del webserver-ca
------------------------
Deleted CA "webserver-ca"
------------------------
```

验证

- 运行 ipa ca-find 来显示 CA 和子 CA 的列表。webserver-ca 不再位于列表中。

```
# ipa ca-find
-------------
2 CAs matched
-------------
  Name: ipa
  Description: IPA CA
  Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
  Subject DN: CN=Certificate Authority,O=IPA.TEST
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

  Name: webclient-ca
  Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
  Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST
----------------------------
Number of entries returned 2
----------------------------
```

## 79.2. 从 IDM WEBUI 下载子 CA 证书

### 先决条件

- 确保您已获取 IdM 管理员的凭据。

### 流程

1. 在 Authentication 菜单中点击 Certificates > Certificates。

图 79.2. 证书列表中的子 CA 证书


2. 单击子 CA 证书的序列号，以打开证书信息页面。

3. 在证书信息页面中，点击 Actions > Download。

4. 在 CLI 中，将子 CA 证书移到 /etc/pki/tls/private/ 目录中：

```
# mv path/to/the/downloaded/certificate /etc/pki/tls/private/sub-ca.crt
```

## 79.3. 为 WEB 服务器和客户端身份验证创建 CA ACL

证书颁发机构访问控制列表(CA ACL)规则定义哪些配置文件可用于向哪些用户、服务或主机发布证书。通过关联配置文件、主体和组，CA ACL 允许主体或组使用特定配置集请求证书。

例如，利用 CA ACL，管理员可以将适用于从伦敦办事处工作的员工的配置文件的使用限制为属于伦敦办事处相关组的成员的用户。

### 79.3.1. 在 IdM CLI 中查看 CA ACL

按照以下流程查看 IdM 部署中提供的证书颁发机构访问控制列表(CA ACL)以及特定 CA ACL 的详情。

**流程**

1. 要查看 IdM 环境中的所有 CA ACL，请输入 ipa caacl-find 命令：

```
$ ipa caacl-find
----------------
1 CA ACL matched
----------------
  ACL name: hosts_services_calPAserviceCert
  Enabled: TRUE
```

2. 要查看 CA ACL 的详细信息，请输入 ipa caacl-show 命令并指定 CA ACL 名称。例如，要查看 hosts_services_calPAserviceCert CA ACL 的详情，请输入：

```
$ ipa caacl-show hosts_services_calPAserviceCert
  ACL name: hosts_services_calPAserviceCert
  Enabled: TRUE
  Host category: all
  Service category: all
  CAs: ipa
  Profiles: calPAserviceCert
  Users: admin
```

### 79.3.2. 为使用 webserver-ca 发布的证书向 Web 客户端进行身份验证的 Web 服务器创建 CA ACL

按照以下流程，在为 HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM 服务请求证书时，创建一个需要系统管理员使用 webserver-ca 子 CA 和 calPAserviceCert 配置文件的 CA ACL。如果用户从其他子 CA 或不同配置集请求证书，则请求会失败。唯一的例外是在启用了另一个匹配的 CA ACL 时。要查看可用的 CA ACL，请参阅在 IdM CLI 中查看 CA ACL。

**先决条件**

- 确保 HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM 服务是 IdM 的一部分。
- 确保您已获取 IdM 管理员的凭据。

**流程**

1. 使用 ipa caacl 命令创建 CA ACL，并指定其名称：

```
$ ipa caacl-add TLS_web_server_authentication
----------------------------------------
Added CA ACL "TLS_web_server_authentication"
----------------------------------------
  ACL name: TLS_web_server_authentication
  Enabled: TRUE
```

2. 使用 ipa caacl-mod 命令修改 CA ACL 以指定 CA ACL 的说明：

```
$ ipa caacl-mod TLS_web_server_authentication --desc="CAACL for web servers
authenticating to web clients using certificates issued by webserver-ca"
----------------------------------------------
Modified CA ACL "TLS_web_server_authentication"
----------------------------------------------
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
  Enabled: TRUE
```

3. 将 webserver-ca 子 CA 添加到 CA ACL 中：

```
$ ipa caacl-add-ca TLS_web_server_authentication --ca=webserver-ca
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
  Enabled: TRUE
  CAs: webserver-ca
------------------------
Number of members added 1
------------------------
```

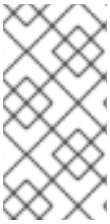
4. 使用 ipa caacl-add-service 指定主体可以请求证书的服务：

```
$ ipa caacl-add-service TLS_web_server_authentication --
service=HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
  Enabled: TRUE
  CAs: webserver-ca
  Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
-------------------------
Number of members added 1
-------------------------
```

5. 使用 ipa caacl-add-profile 命令为请求的证书指定证书配置集：

```
$ ipa caacl-add-profile TLS_web_server_authentication --
certprofiles=caIPAserviceCert
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
  Enabled: TRUE
  CAs: webserver-ca
  Profiles: caIPAserviceCert
  Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
-------------------------
Number of members added 1
-------------------------
```

您可以直接使用新创建的 CA ACL。它在创建后默认启用。



注意

CA ACL 的重点是指定允许哪些 CA 和配置文件组合用于来自特定主体或组的请求。CA ACL 不会影响证书验证或信任。它们不会影响签发的证书的使用方式。

### 79.3.3. 为用户 Web 浏览器创建 CA ACL，使用 webclient-ca 发布的证书向 Web 服务器进行身份验证

按照以下流程，在请求证书时，创建一个需要系统管理员使用 webclient-ca 子 CA 和 IECUserRoles 配置文件的 CA ACL。如果用户从其他子 CA 或不同配置集请求证书，则请求会失败。唯一的例外是在启用了另一个匹配的 CA ACL 时。要查看可用的 CA ACL，请参阅在 IdM CLI 中查看 CA ACL。

**先决条件**

- 确保您已获取 IdM 管理员的凭据。

**流程**

1. 使用 ipa caacl 命令创建 CA ACL 并指定其名称：

```
$ ipa caacl-add TLS_web_client_authentication
---------------------------------------------
Added CA ACL "TLS_web_client_authentication"
---------------------------------------------
  ACL name: TLS_web_client_authentication
  Enabled: TRUE
```

2. 使用 ipa caacl-mod 命令修改 CA ACL 以指定 CA ACL 的说明：

```
$ ipa caacl-mod TLS_web_client_authentication --desc="CAACL for user web
browsers authenticating to web servers using certificates issued by webclient-ca"
-----------------------------------------------
Modified CA ACL "TLS_web_client_authentication"
-----------------------------------------------
  ACL name: TLS_web_client_authentication
  Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
  Enabled: TRUE
```

3. 将 webclient-ca 子 CA 添加到 CA ACL 中：

```
$ ipa caacl-add-ca TLS_web_client_authentication --ca=webclient-ca
  ACL name: TLS_web_client_authentication
  Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
  Enabled: TRUE
  CAs: webclient-ca
------------------------
Number of members added 1
------------------------
```

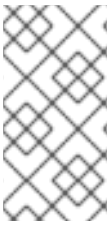
4. 使用 ipa caacl-add-profile 命令为请求的证书指定证书配置集：

```
$ ipa caacl-add-profile TLS_web_client_authentication --certprofiles=IECUserRoles
  ACL name: TLS_web_client_authentication
  Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
  Enabled: TRUE
  CAs: webclient-ca
  Profiles: IECUserRoles
------------------------
Number of members added 1
------------------------
```

5. 使用 ipa caacl-mod 命令修改 CA ACL，以指定 CA ACL 适用于所有 IdM 用户：

```
$ ipa caacl-mod TLS_web_client_authentication --usercat=all
----------------------------------------------
Modified CA ACL "TLS_web_client_authentication"
----------------------------------------------
  ACL name: TLS_web_client_authentication
  Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
  Enabled: TRUE
  User category: all
  CAs: webclient-ca
  Profiles: IECUserRoles
```

您可以直接使用新创建的 CA ACL。它在创建后默认启用。



注意

CA ACL 的重点是指定允许哪些 CA 和配置文件组合用于来自特定主体或组的请求。CA ACL 不会影响证书验证或信任。它们不会影响签发的证书的使用方式。

## 79.4. 使用 CERTMONGER 为服务获取 IDM 证书

为确保浏览器和在 IdM 客户端上运行的 Web 服务之间的通信安全且加密，请使用 TLS 证书。如果要将 Web 浏览器限制为信任 webserver-ca 子 CA 发布但没有其它 IdM 子 CA 发布的证书，请从 webserver-ca 子 CA 获取 Web 服务的 TLS 证书。

按照以下流程，使用 certmonger 获取在 IdM 客户端上运行的服务 (HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM)的 IdM 证书。

使用 证书监控 器自动请求证书意味着，certmonger 在到期需要续订时管理和续订证书。

有关 certmonger 请求服务证书时发生的情况的可视化表示，请参阅 请求服务证书的 certmonger 的通信流。

先决条件

- Web 服务器已注册为 IdM 客户端。

- 您有正在运行的 IdM 客户端的 root 访问权限。
- 请求证书的服务不必在 IdM 中预先存在。

**流程**

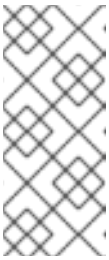
1. 在运行 HTTP 服务的 my_company.idm.example.com IdM 客户端中，请求与 HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM 主体对应的服务的证书，并指定：
   - 证书将存储在本地 /etc/pki/tls/certs/httpd.pem 文件中
   - 私钥存储在本地 /etc/pki/tls/private/httpd.key 文件中
   - The webserver-ca 子 CA 将作为发行证书颁发机构
   - 将 SubjectAltName 的 extensionRequest 添加到签名请求中，其 DNS 名称为 my_company.idm.example.com ：

```
# ipa-getcert request -K HTTP/my_company.idm.example.com -k
/etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
my_company.idm.example.com -X webserver-ca -C "systemctl restart httpd"
New signing request "20190604065735" added.
```

在以上命令中：

- ipa-getcert request 命令指定要从 IdM CA 获取证书。ipa-getcert request 命令是 getcert request -c IPA 的快捷方式。
- g 选项指定要生成的密钥的大小（如果尚未到位）。
- D 选项指定 要添加到请求的 SubjectAltName DNS 值。

- X 选项指定 证书的签发者必须是 webserver-ca， 而不是 ipa。
- C 选项 指示 certmonger 在获取证书后重新启动 httpd 服务。
- 要指定证书与特定的配置集一起发布，请使用 -T 选项。



**注意**

RHEL 8 在 Apache 中使用与 RHEL 7 中使用的不同的 SSL 模块。SSL 模块依赖于 OpenSSL 而不是 NSS。因此，在 RHEL 8 中，您无法使用 NSS 数据库存储 HTTPS 证书和私钥。

2. （可选）检查请求的状态：

```
# ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
   status: MONITORING
   stuck: no
   key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
   certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
   CA: IPA
   issuer: CN=WEBSERVER,O=IDM.EXAMPLE.COM

[...]
```

输出显示请求处于 MONITORING 状态，这表示已获取了证书。密钥对和证书的位置是请求的位置。

## 79.5. 请求服务证书的证书的通信流

这些图显示了当 certmonger 从身份管理(IdM)证书认证机构(CA)服务器请求服务证书时发生了什么情况的阶段。序列由这些图表组成：

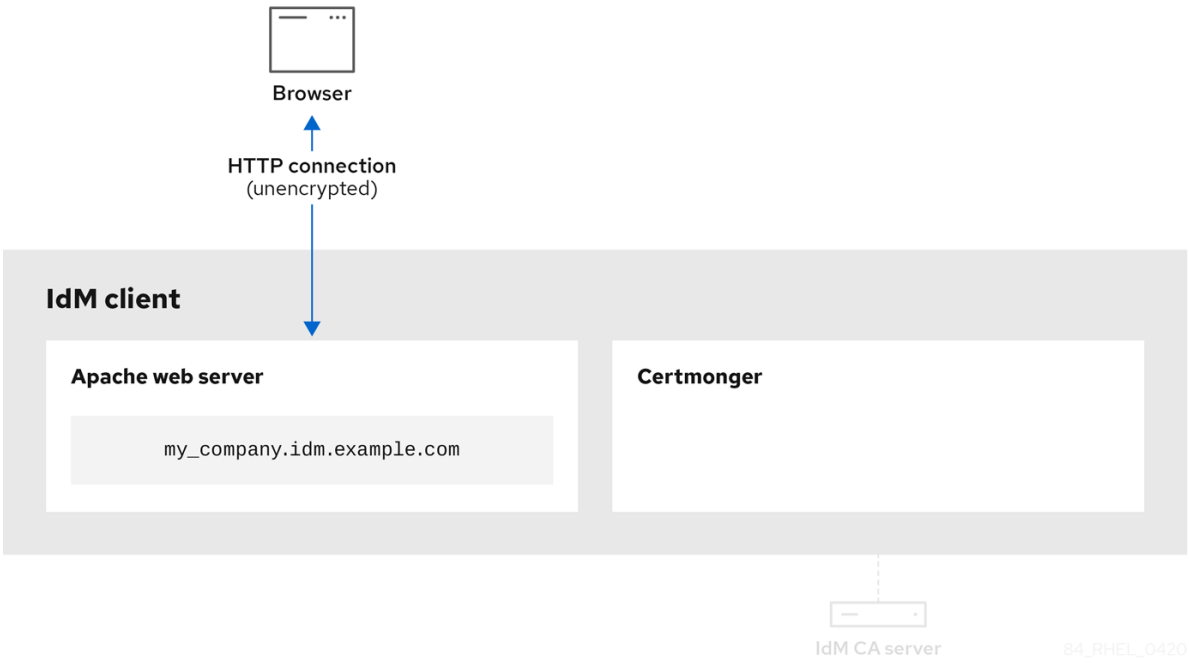
- 未加密的通信
- 请求服务证书的 certmonger

- 发布服务证书的 IdM CA
- 应用服务证书的 certmonger
- 当旧的证书接近过期时，请求新证书的 certmonger

在图中，webserver-ca 子 CA 由通用 IdM CA 服务器 表示。

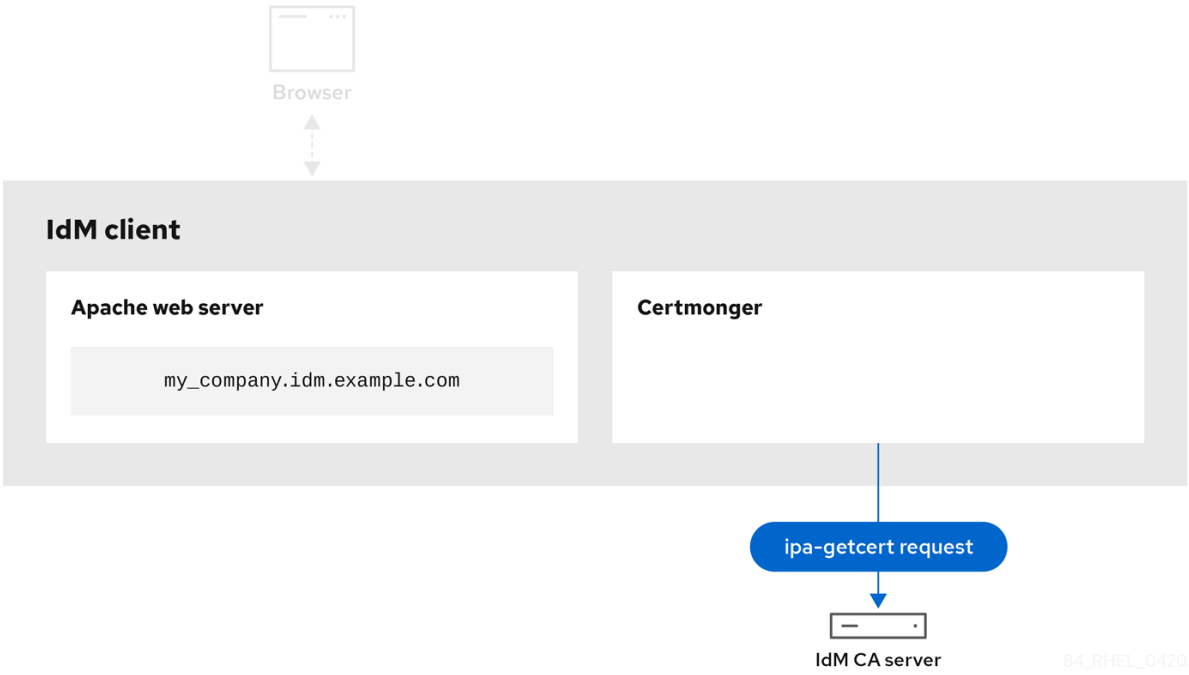
未加密的通信 显示初始情况：没有 HTTPS 证书，Web 服务器和浏览器之间的通信是未加密的。

图 79.3. 未加密的通信



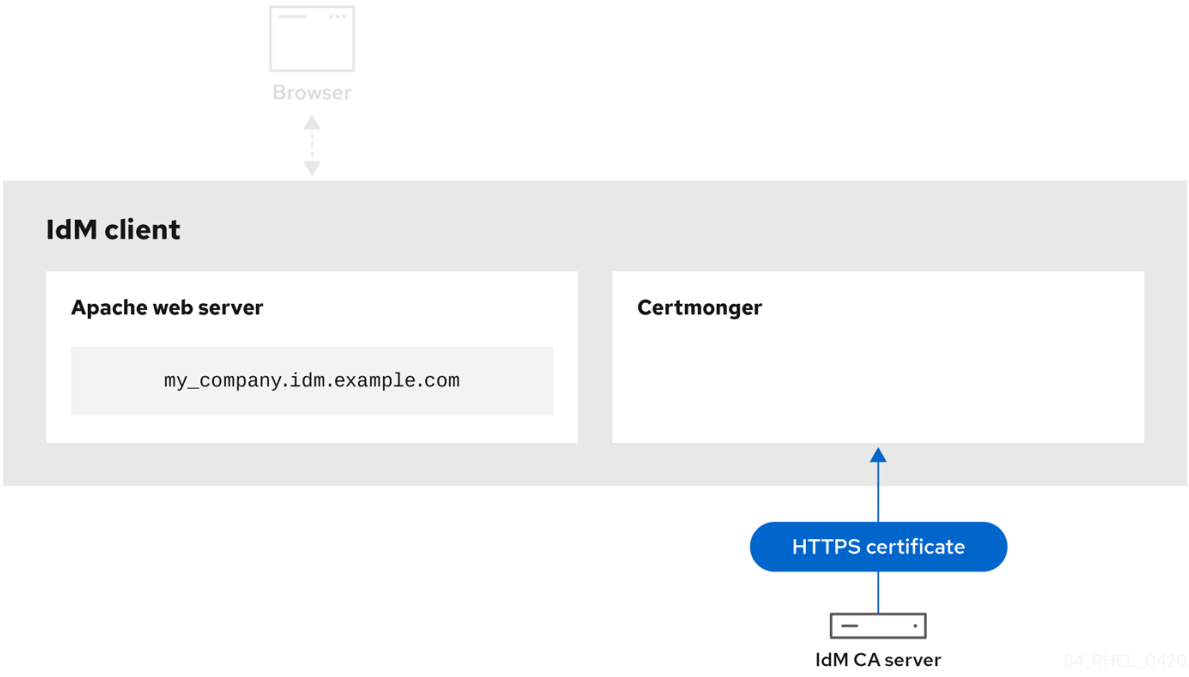

请求服务证书的 certmonger 显示系统管理员使用 certmonger 来手动为 Apache Web 服务器请求 HTTPS 证书。请注意，在请求 Web 服务器证书时，certmonger 不会直接与 CA 通信。它通过 IdM 代理。

图 79.4. 请求服务证书的 certmonger



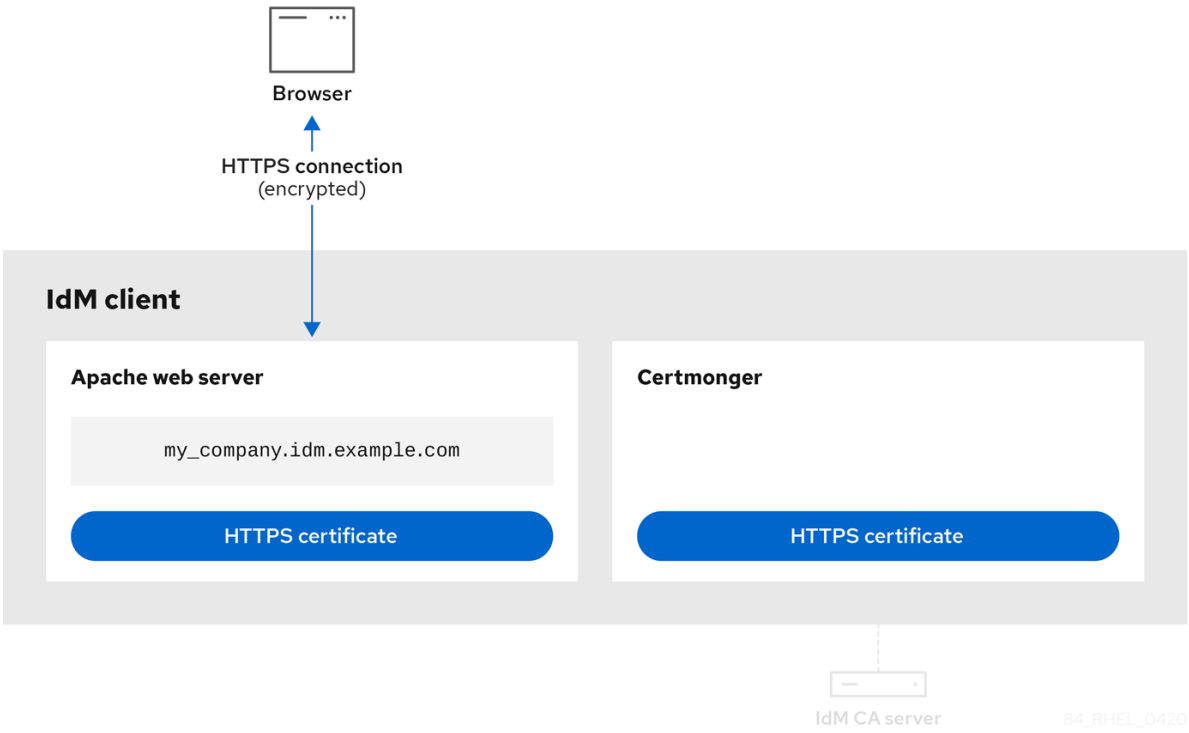

发布服务证书的 IdM CA 显示为 web 服务器发出 HTTPS 证书的 IdM CA。

图 79.5. 发布服务证书的 IdM CA



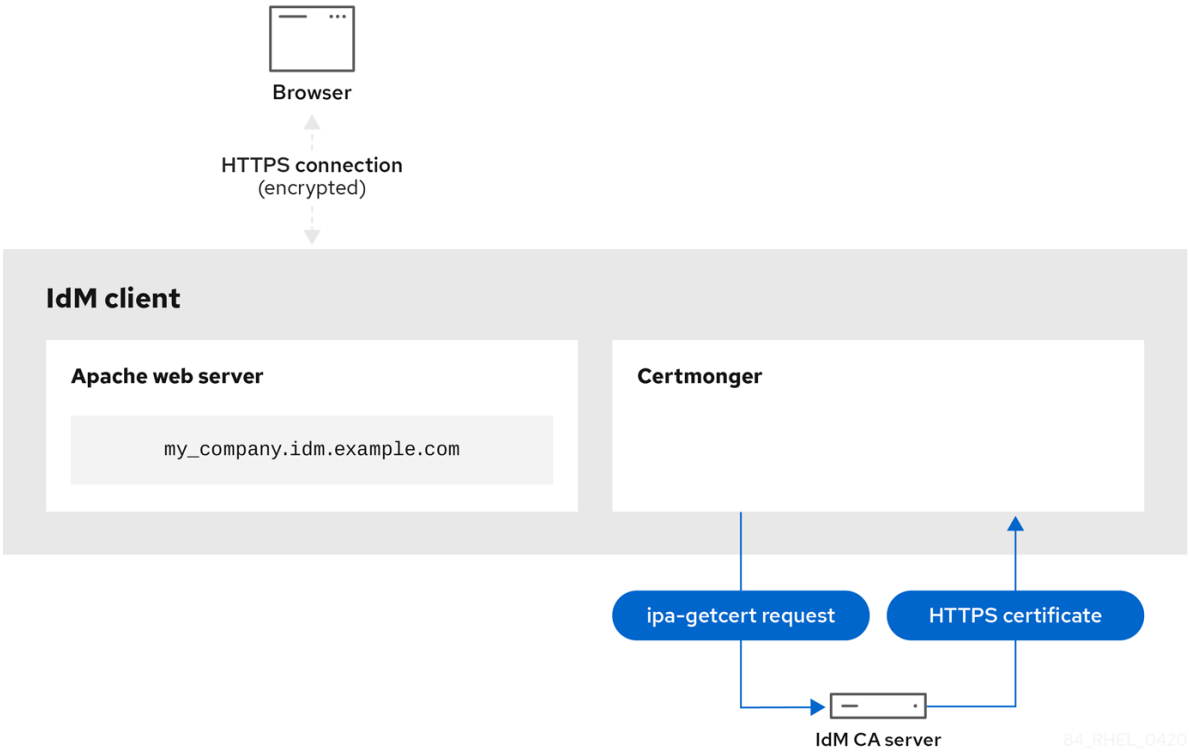
应用服务证书的 certmonger 显示将 HTTPS 证书放在 IdM 客户端上合适位置的 certmonger，如果指示要这样做，请重新启动 httpd 服务。随后 Apache 服务器使用 HTTPS 证书来加密自身和浏览器之间的流量。

图 79.6. 应用服务证书的 certmonger




当旧证书接近过期时，请求新证书的 certmonger，显示 certmonger 在证书过期前自动从 IdM CA 请求续订服务证书。IdM CA 发布一个新证书。

图 79.7. 当旧的证书接近过期时，请求新证书的 certmonger




## 79.6. 设置单实例 APACHE HTTP 服务器

您可以设置一个单实例 Apache HTTP 服务器来提供静态 HTML 内容。

如果 Web 服务器应该为与服务器关联的所有域提供相同的内容，请按照流程操作。如果要为不同的域提供不同的内容，请设置基于名称的虚拟主机。详情请参阅配置基于 Apache 名称的虚拟主机。

**流程**

1. 安装 httpd 软件包：

```
# yum install httpd
```

2. 如果使用 firewalld，请在本地防火墙中打开 TCP 端口 80 ：

```
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --reload
```

3. 启用并启动 httpd 服务：

```
# systemctl enable --now httpd
```

4. 可选：将 HTML 文件添加到 /var/www/html/ 目录中。

**注意**

在 向/var/www/html/ 添加内容时，在httpd默认运行的情况下，文件和目录必须可被用户读取。内容所有者可以是 root用户和root用户组，也可以是管理员所选择的其他用户或组。如果内容所有者是 root 用户和 root 用户组，则文件必须可被其他用户读取。所有文件和目录的 SELinux 上下文必须为 httpd_sys_content_t，其默认应用于 /var/www 目录中的所有内容。

验证步骤

- 使用 Web 浏览器连接到 http://my_company.idm.example.com/ 或 http://*server_IP*/。

如果 /var/www/html/ 目录为空，或者不包含 index.html或index.htm文件，则 Apache 会显示 Red Hat Enterprise Linux 测试页面。如果 /var/www/html/ 包含具有不同名称的 HTML 文件，您可以通过在该文件中输入 URL 来加载这些文件，如 http://*server_IP*/*example.html* 或 http://my_company.idm.example.com/*example.html*。

其它资源

- Apache 手册： 安装 Apache HTTP 服务器手册。
- 请参见 httpd.service(8) 手册页。

### 79.7. 在 APACHE HTTP 服务器中添加 TLS 加密

您可以对 idm.example.com 域的 my_company.idm.example.com Apache HTTP 服务器启用 TLS 加密。

先决条件

- my_company.idm.example.com Apache HTTP 服务器已安装并在运行。

- 您已从 webserver-ca 子 CA 获取 TLS 证书，并将其存储在 /etc/pki/tls/certs/httpd.pem 文件中，如使用 certmonger 的服务的 Obtaining IdM 证书 所述。如果您使用其他路径，请调整该流程的对应步骤。

- 对应的私钥存储在 /etc/pki/tls/private/httpd.key 文件中。如果您使用其他路径，请调整该流程的对应步骤。

- webserver-ca CA 证书存储在 /etc/pki/tls/private/sub-ca.crt 文件中。如果您使用其他路径，请调整该流程的对应步骤。

- 客户端和 my_company.idm.example.com Web 服务器将服务器的主机名解析为 Web 服务器的 IP 地址。

流程

1. 安装 mod_ssl 软件包：

```
# yum install mod_ssl
```

2. 编辑/etc/httpd/conf.d/ssl.conf文件，并将以下设置添加到 <VirtualHost _default_:443>指令中：

   a. 设置服务器名称：

```
ServerName my_company.idm.example.com
```

重要

服务器名称必须与证书的 Common Name字段中设置的条目匹配。

   a. 可选：如果证书在 Subject Alt Names (SAN)字段中包含额外的主机名，您可以 配置mod_ssl来为这些主机名提供 TLS 加密。要配置此功能，请添加具有对应名称的ServerAliases参数：

```
ServerAlias www.my_company.idm.example.com
server.my_company.idm.example.com
```

b. 设置到私钥、服务器证书和 CA 证书的路径：

```
SSLCertificateKeyFile "/etc/pki/tls/private/httpd.key"
SSLCertificateFile "/etc/pki/tls/certs/httpd.pem"
SSLCACertificateFile "/etc/pki/tls/certs/ca.crt"
```

3. 出于安全考虑，配置成只有 root 用户才可以访问私钥文件：

```
# chown root:root /etc/pki/tls/private/httpd.key
# chmod 600 //etc/pki/tls/private/httpd.key
```

**警告**

如果私钥被设置为可以被未授权的用户访问，则需要撤销证书，然后再创建一个新私钥并请求一个新证书。否则，TLS 连接就不再安全。

4. 如果您使用 firewalld，在本地防火墙中打开端口 443 ：

```
# firewall-cmd --permanent --add-port=443/tcp
# firewall-cmd --reload
```

5. 重启httpd服务：

```
# systemctl restart httpd
```

**注意**

如果您使用密码来保护私钥文件，则必须在每次 httpd服务启动时都输入此密码。

- 使用浏览器并连接到 https://my_company.idm.example.com 。

**其它资源**

- SSL/TLS 加密.
- RHEL 8 中 TLS 的安全注意事项

## 79.8. 在 APACHE HTTP 服务器中设置支持的 TLS 协议版本

默认情况下，RHEL 上的 Apache HTTP 服务器使用定义了安全默认值的系统范围的加密策略，这些值也与最新的浏览器兼容。例如，DEFAULT策略定义了在 apache 中只启用 TLSv1.2和TLSv1.3协议版本。

您可以手动配置 my_company.idm.example.com Apache HTTP 服务器支持哪些 TLS 协议版本。如果您的环境只需要启用特定的 TLS 协议版本，请按照以下步骤操作，例如：

- 如果您的环境要求客户端也可以使用弱 TLS1 (TLSv1.0)或TLS1.1协议。
- 如果你想将 Apache 配置为只支持TLSv1.2或TLSv1.3协议。

先决条件

- 在 my_company.idm.example.com 服务器上启用 TLS 加密，如 向 Apache HTTP 服务器添加 TLS 加密 所述。

流程

1. 编辑 /etc/httpd/conf/httpd.conf 文件，并将以下设置添加到您要为其设置 TLS 协议版本的<VirtualHost>指令中。例如，只启用TLSv1.3协议：

```
SSLProtocol -All TLSv1.3
```

2. 重启httpd服务：

```
# systemctl restart httpd
```

验证步骤

1. 使用以下命令来验证服务器是否支持TLSv1.3:

```
# openssl s_client -connect example.com:443 -tls1_3
```

2. 使用以下命令来验证服务器是否不支持TLSv1.2 ：

```
# openssl s_client -connect example.com:443 -tls1_2
```

如果服务器不支持该协议，命令会返回一个错误：

```
140111600609088:error:1409442E:SSL routines:ssl3_read_bytes:tlsv1 alert protocol
version:ssl/record/rec_layer_s3.c:1543:SSL alert number 70
```

3. 可选：重复用于其他 TLS 协议版本的命令。

**其它资源**

- update-crypto-policies(8) 手册页
- 使用系统范围的加密策略。
- 有关 SSLProtocol 参数的详情，请参考 Apache 手册中的 mod_ssl 文档： 安装 Apache HTTP 服务器手册。

### 79.9. 在 APACHE HTTP 服务器中设置支持的密码

默认情况下， Apache HTTP 服务器使用定义了安全默认值的系统范围的加密策略，这些值也与最新的浏览器兼容。有关系统范围加密允许的密码列表，请查看/etc/crypto-policies/back-ends/openssl.config 文件。

您可以手动配置 my_company.idm.example.com Apache HTTP 服务器支持哪些密码。如果您的环境需要特定的加密系统，请按照以下步骤操作。

**先决条件**

-

在 my_company.idm.example.com 服务器上启用 TLS 加密，如 向 Apache HTTP 服务器添加 TLS 加密 所述。

流程

1. 编辑/etc/httpd/conf/httpd.conf文件，并将SSLCipherSuite参数添加到您要为其设置 TLS 密码的<VirtualHost>指令中：

```
SSLCipherSuite
"EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH:!SHA1:!SHA256"
```

这个示例只启用 EECDH+AESGCM、EDH+AESGCM、AES256+EECDH 和 AES256+EDH密码，并禁用所有使用SHA1和SHA256消息身份验证码(MAC)的密码。

2. 重启httpd服务：

```
# systemctl restart httpd
```

验证步骤

1. 显示 Apache HTTP 服务器支持的密码列表：

a. 安装nmap软件包：

```
# yum install nmap
```

b. 使用nmap工具来显示支持的加密：

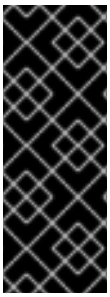
```
# nmap --script ssl-enum-ciphers -p 443 example.com
...
PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
...
```

其它资源

- update-crypto-policies(8) 手册页
- 使用系统范围的加密策略。
- 安装 Apache HTTP 服务器手册 - SSLCipherSuite

### 79.10. 配置 TLS 客户端证书身份验证

客户端证书身份验证仅允许使用证书进行身份验证的用户访问 my_company.idm.example.com Web 服务器上的资源。您可以为 /var/www/html/Example/ 目录配置客户端证书身份验证。



重要

如果 my_company.idm.example.com Apache 服务器使用 TLS 1.3 协议，则某些客户端需要额外的配置。例如，在 Firefox 中，将about:config菜单中的security.tls.enable_post_handshake_auth参数设置为true。详情请查看 Red Hat Enterprise Linux 8中的传输层安全版本1.3。

**先决条件**

- 在 my_company.idm.example.com 服务器上启用 TLS 加密，如 向 Apache HTTP 服务器添加 TLS 加密 所述。

**流程**

1. 编辑/etc/httpd/conf/httpd.conf文件，并将以下设置添加到你要为其配置客户端验证的<VirtualHost>指令中：

```
<Directory "/var/www/html/Example/">
  SSLVerifyClient require
</Directory>
```

SSLVerifyClient require设置定义了服务器必须成功验证客户端证书，然后客户端才能访问/var/www/html/Example/目录中的内容。

2. 重启httpd服务：

```
# systemctl restart httpd
```

验证步骤

1. 使用 curl 工具在没有客户端身份验证的情况下访问 https://my_company.idm.example.com/Example/ URL：

```
$ curl https://my_company.idm.example.com/Example/
curl: (56) OpenSSL SSL_read: error:1409445C:SSL routines:ssl3_read_bytes:tlsv13 alert
certificate required, errno 0
```

此错误表示 my_company.idm.example.com Web 服务器需要客户端证书身份验证。

2. 将客户端私钥和证书以及 CA 证书传递给curl以便使用客户端身份验证来访问相同的URL：

```
$ curl --cacert ca.crt --key client.key --cert client.crt
https://my_company.idm.example.com/Example/
```

如果请求成功，curl会显示存储在/var/www/html/Example/目录中的index.html文件。

其它资源

- 安装 Apache HTTP 服务器手册 - mod_ssl 配置

## 79.11. 请求新的用户证书并将其导出到客户端

作为身份管理(IdM)管理员，您可以配置在 IdM 客户端上运行的 Web 服务器，以请求使用 Web 浏览器访问服务器的用户对特定 IdM 子 CA 发布的证书进行身份验证。按照以下流程，从特定的 IdM 子 CA 请求用户证书，并将主机上的证书和对应的私钥导出到用户希望使用 Web 浏览器访问 Web 服务器的主机上。之后，将证书和私钥导入到浏览器中。

流程

1. （可选）创建新目录，如 ~/certdb/，并使其成为临时证书数据库。当系统提示时，创建一个 NSS 证书数据库密码来加密后续步骤中生成的证书的密钥：

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
```

```
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2. 创建证书签名请求(CSR)，并将输出重定向到文件。例如，要为 IDM.EXAMPLE.COM 域中的 idm_user 用户创建一个名称为 certificate_request.csr 的 4096 位 CSR，请将证书私钥的昵称设为 idm_user 以便于查找，并将主题设为 CN=idm_user,O=IDM.EXAMPLE.COM ：

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s
"CN=idm_user,O=IDM.EXAMPLE.COM" > certificate_request.csr
```

3. 出现提示时，输入您在使用 certutil 创建临时数据库时输入的相同密码。然后继续键入 rundlomly 直到通知停止：

```
Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the
creation of your key.  One of the easiest ways to create a
random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter
is full.  DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!


Continue typing until the progress meter is full:
```

4. 将证书请求文件提交到服务器。指定要与新发布的证书关联的 Kerberos 主体、存储证书的输出文件，以及可选的证书配置集。指定您要签发证书的 IdM 子 CA。例如，要获取 IECUserRoles 配置集的证书，带有添加的用户角色扩展的配置文件，idm_user@IDM.EXAMPLE.COM 主体来自 webclient-ca，并将证书保存到 ~/idm_user.pem 文件中：

```
# ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --ca=webclient-ca --certificate-out=~/idm_user.pem
```

5. 将证书添加到 NSS 数据库。使用 -n 选项设置之前创建 CSR 时所用的相同 nickname，以便该证书与 NSS 数据库中的私钥相匹配。t 选项设置信任级别。详情请查看 certutil(1)man page。i 选项指定 输入证书文件。例如，要将一个具有 idm_user 昵称的证书添加到 NSS 数据库中，该证书存储在 ~/certdb/ 数据库中的 ~/idm_user.pem 文件中：

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

6. 验证 NSS 数据库中的密钥是否未显示 （或称为） 作为其 nickname。例如，验证存储在 ~/certdb/ 数据库中 的证书没有被孤立：

```
# certutil -K -d ~/certdb/
< 0> rsa      5ad14d41463b87a095b1896cf0068ccc467df395   NSS Certificate
DB:idm_user
```

7. 使用 pk12util 命令将证书从 NSS 数据库导出到 PKCS12 格式。例如，将 /root/certdb NSS 数据库中的 idm_user nickname 的证书导出到 ~/idm_user.p12 文件：

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

8. 将证书传输到您要启用 idm_user 的证书身份验证的主机：

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

9. 在传输了证书的主机上，出于安全原因使 'pkcs12 文件被 'other' 组无法访问的目录：

```
# chmod o-rwx /home/idm_user/
```

10. 出于安全考虑，请从服务器中删除临时 NSS 数据库和 .pkcs12 文件：

```
# rm ~/certdb/
# rm ~/idm_user.p12
```

### 79.12. 配置浏览器以启用证书身份验证

若要在使用 Web UI 登录身份管理(IdM)时通过证书进行身份验证，您需要将用户和相关证书颁发机构(CA)证书导入到 Mozilla Firefox 或 Google Chrome 浏览器。浏览器运行的主机本身不必是 IdM 域的一部分。

IdM 支持以下浏览器来连接到 WebUI：

- Mozilla Firefox 38 及更新的版本

- Google Chrome 46 及更新的版本

以下流程演示了如何配置 Mozilla Firefox 57.0.1 浏览器。

**先决条件**

- 您可以使用 PKCS#12 格式在浏览器中导入 用户证书。
- 您已 下载子 CA 证书，并以 PEM 格式随时可用。

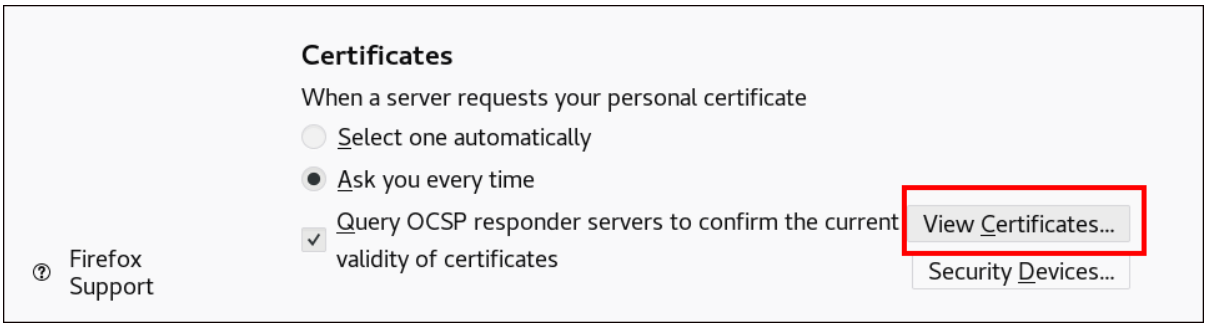
**流程**

1. 打开 Firefox，然后导航到 Preferences → Privacy & Security。

图 79.8. Preferences 中的隐私和安全部分


2. 单击 查看证书。

图 79.9. 查看隐私和安全性中的证书




3. 在您的 证书 选项卡中，单击 Import。以 PKCS12 格式查找并打开用户证书，然后点 OK 和

OK。

4. 要确保您的 IdM 子 CA 被 Firefox 识别为可信颁发机构，请导入您在 从 IdM Web UI 下载子 CA 证书 中作为可信证书颁发机构证书保存的证书：

a. 打开 Firefox，导航到 Preferences 并点击 Privacy & Security。

图 79.10. Preferences 中的隐私和安全部分


b. 单击 查看证书。

图 79.11. 查看隐私和安全性中的证书


c. 在" 颁发机构 "选项卡中，单击 Import。查找并打开子 CA 证书。信任证书来识别网站，然后点OK 和 OK。

# 第 80 章 快速使特定一组相关证书无效

作为系统管理员，如果您希望快速使一组特定证书无效：

- 设计您的应用，以便他们仅信任由特定轻量身份管理(IdM)子 CA 发布的证书。之后，您只需撤销签发这些证书的 Identity Management(IdM)子 CA 的证书，即可使所有这些证书无效。有关如何在 IdM 中创建和使用轻量级子 CA 的详情，请参考 快速地使特定的相关证书组无效。

- 为确保正在取消的 IdM 子 CA 发布的所有证书都立即无效，请配置依赖此类证书的应用程序使用 IdM OCSP 响应者。例如，若要将 Firefox 浏览器配置为使用 OCSP 响应器，请确保在 Firefox Preferences 中选中了 查询 OCSP 响应器服务器以确认证书复选框当前有效。

  在 IdM 中，证书吊销列表(CRL)每四个小时更新一次。要使 IdM 子 CA 发布的所有证书无效，请参阅 吊销 IdM 子 CA 证书。此外，禁用相关的 CA ACL，并考虑 禁用 IdM 子 CA。禁用子 CA 可防止子 CA 发布新证书，但允许为之前发布的证书生成在线证书状态协议(OCSP)响应，因为子 CA 的签名密钥被保留。

重要

如果您的环境中使用 OCSP，则不要删除子 CA。删除子 CA 会删除子 CA 的签名密钥，从而导致为该子 CA 发布的证书生成 OCSP 响应。

删除子 CA 时的唯一场景是希望创建一个具有相同主题区分名称(DN)但新的签名密钥的新子 CA，而不是禁用它。

## 80.1. 在 IDM CLI 中禁用 CA ACL

当您要停用 IdM 服务或一组 IdM 服务时，请考虑禁用任何现有的相应 CA ACL。

按照以下流程禁用 TLS_web_server_authentication CA ACL，其限制运行在 IdM 客户端上的 Web 服务器请求由 webserver-ca IdM 子 CA 发布的证书，并禁用 TLS_web_client_authentication CA ACL，其限制 IdM 用户请求由 webclient-ca IdM 子 CA 发布的用户证书。

**流程**

1. 要查看 IdM 环境中的所有 CA ACL，请输入 ipa caacl-find 命令：

```
$ ipa caacl-find
----------------
3 CA ACLs matched
----------------
  ACL name: hosts_services_calPAserviceCert
  Enabled: TRUE

  ACL name: TLS_web_server_authentication
  Enabled: TRUE

  ACL name: TLS_web_client_authentication
  Enabled: TRUE
```

2. 另外，若要查看 CA ACL 的详情，请输入 ipa caacl-show 命令并指定 CA ACL 名称：

```
$ ipa caacl-show TLS_web_server_authentication
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
  Enabled: TRUE
  CAs: webserver-ca
  Profiles: calPAserviceCert
  Services: HTTP/rhel8server.idm.example.com@IDM.EXAMPLE.COM
```

3. 要禁用 CA ACL，输入 ipa caacl-disable 命令并指定 CA ACL 名称。

- 要禁用 TLS_web_server_authentication CA ACL，请输入：

```
$ ipa caacl-disable TLS_web_server_authentication
-----------------------------------------------
Disabled CA ACL "TLS_web_server_authentication"
-----------------------------------------------
```

- 要禁用 TLS_web_client_authentication CA ACL，请输入：

```
$ ipa caacl-disable TLS_web_client_authentication
-----------------------------------------------
Disabled CA ACL "TLS_web_client_authentication"
-----------------------------------------------
```

现在唯一启用的 CA ACL 是 hosts_services_calPAserviceCert CA ACL。

**重要**

对于禁用 hosts_services_caIPAserviceCert CA ACL，请格外小心。禁用 hosts_services_caIPAserviceCert，没有另一个 CA ACL 允许 IdM 服务器使用带有 caIPAserviceCert 配置集的 ipa CA，这意味着 IdM HTTP 和 LDAP 证书的证书续订会失败。过期的 IdM HTTP 和 LDAP 证书最终将导致 IdM 系统失败。

## 80.2. 禁用 IDM 子 CA

在撤销 IdM 子 CA 的 CA 证书以使该子 CA 发布的所有证书无效后，如果您不再需要 IdM 子 CA，请考虑禁用它。您可以稍后重新启用子 CA。

禁用子 CA 可防止子 CA 发布新证书，但允许为之前发布的证书生成在线证书状态协议(OCSP)响应，因为子 CA 的签名密钥被保留。

**先决条件**

- 以 IdM 管理员身份登录。

**流程**

- 输入 ipa ca-disable 命令并指定 sub-CA 的名称：

```
$ ipa ca-disable webserver-CA
--------------------
Disabled CA "webserver-CA"
--------------------
```

# 第 81 章 IDM 中的 VAULTS

本章论述了身份管理(IdM)中的密码库。它包括以下主题：

- 库的概念.
- 与密码库关联的不同角色。
- IdM 中根据安全性和访问控制级别提供的不同库类型。
- IdM 中根据所有权可用的不同类型的库。
- vault 容器的概念.
- 在 IdM 中管理密码库的基本命令。
- 安装密钥恢复颁发机构(KRA)，这是在 IdM 中使用密码库的先决条件。

## 81.1. VAULT 及其益处

对于希望将其所有敏感数据保存在一个位置，其身份管理(IdM)用户而言，密码库是一种非常有用的功能。 有各种类型的 vault，您应该根据您的要求选择要使用的 vault。

密码库在(IdM)中是一个安全的位置，用于存储、检索、共享和恢复机密。secret 是安全敏感数据，通常是身份验证凭据，只有有限的人员或实体可以访问这些数据。例如，secret 包括：

- 密码
- pins

- 私有 SSH 密钥

密码库与密码管理器相当。正如密码管理器一样，密码库通常要求用户生成并记住一个主密码来解锁和访问密码库中存储的任何信息。但是，用户也可以决定拥有标准密码库。标准密码库不要求用户输入任何密码来访问密码库中存储的 secret。

注意

IdM 中的密码库的目的是存储身份验证凭据，允许您向外部的非IdM 相关服务进行身份验证。

IdM 库的其他重要特征包括：

- Vault 只能供 vault 所有者和 vault 所有者选择为 vault 成员的 IdM 用户访问。此外，IdM 管理员还可以访问密码库。
- 如果用户没有足够的特权来创建密码库，IdM 管理员可以创建密码库并将该用户设置为其所有者。
- 用户和服务可从 IdM 域注册的任何机器访问存储在密码库中的 secret。
- 个密码库只能包含一个机密，例如一个文件。但是，文件本身可以包含多个机密，如密码、密钥选项卡或证书。

注意

Vault 仅在 IdM 命令行(CLI)中可用，不能来自 IdM Web UI。

## 81.2. VAULT 所有者、成员和管理员

身份管理(IdM)区分以下 vault 用户类型：

Vault 所有者

vault 所有者是具有密码库基本管理权限的用户或服务。例如，密码库所有者可以修改密码库的属性或添加新的 vault 成员。

每个密码库必须至少有一个所有者。库也可以有多个所有者。

### Vault 成员

vault 成员是可以访问由其他用户或服务创建的库的用户或服务。

### Vault 管理员

Vault 管理员不受限制地访问所有密码库，并有权执行所有密码库操作。

**注意**

对称和非对称库通过密码或密钥进行保护，并应用特殊的访问控制规则（请参阅 Vault 类型）。管理员必须满足这些规则才能：

- 访问对称和非对称密码库中的机密。
- 更改或重置 vault 密码或密钥。

Vault 管理员是具有 Vault 管理员特权的任何用户。在 IdM 中的基于角色的访问控制(RBAC)的上下文中，特权是您可以应用到角色的一组权限。

### Vault 用户

vault 用户代表密码库所在的容器的用户。Vault 用户信息 显示在特定命令的输出中，如 ipa vault-show ：

```
$ ipa vault-show my_vault
  Vault name: my_vault
  Type: standard
  Owner users: user
  Vault user: user
```

有关 vault 容器和用户密码库的详情，请参阅 Vault 容器。

**其它资源**

- 如需有关 vault 类型的详情，请参阅 标准的、对称的和非对称的vault 。

## 81.3. 标准、对称和非对称密码库

根据安全性和访问控制级别，IdM 将密码库分类为以下类型：

标准密码库

Vault 所有者和密码库成员可以归档和检索机密，而不必使用密码或密钥。

对称密码库

库中的机密通过对称密钥进行保护。Vault 所有者和成员可以归档和检索机密，但必须提供 vault 密码。

非对称密码库

库中的机密通过非对称密钥进行保护。用户使用公钥存档机密，并使用私钥检索该机密。Vault 成员只能存档机密，而 vault 所有者则可同时执行归档和检索机密。

## 81.4. 用户、服务和共享密码库

根据所有权，IdM 将密码库分为几种类型。下表 包含有关每种类型、其所有者和使用的信息。

**表 81.1. 基于所有权的 IdM Vault**

| 类型 | 描述 | 所有者 | 备注 |
| --- | --- | --- | --- |
| **用户密码库** | 用户的私有库 | 单个用户 | 如果 IdM 管理员允许，任何用户都可以拥有一个或多个用户库 |
| **服务库** | 服务的私有库 | 单个服务 | 如果 IdM 管理员允许，任何服务都可以拥有一个或多个用户库 |
| **共享 vault** | 由多个用户和服务共享的库 | 创建密码库的 vault 管理员 | 如果 IdM 管理员允许，用户和服务可以拥有一个或多个用户库。除创建密码库的 vault 管理员之外，还可具有对密码库的完全访问权限。 |

## 81.5. VAULT 容器

vault 容器是密码库的集合。下表列出了 Identity Management (IdM) 提供的默认 vault 容器。

表 81.2. IdM 中的默认 vault 容器

| 类型 | 描述 | 目的 |
| --- | --- | --- |
| 用户容器 | 用户的私有容器 | 为特定用户存储用户密码库 |
| 服务容器 | 服务的私有容器 | 为特定服务存储服务库 |
| 共享容器 | 用于多个用户和服务的容器 | 存储可由多个用户或服务共享的 vault |

当为用户或服务创建第一个私有密码库时，IdM 会自动为每个用户或服务创建用户和服务容器。删除用户或服务后，IdM 会删除容器及其内容。

## 81.6. 基本 IDM VAULT 命令

您可以使用以下介绍的基本命令管理身份管理(IdM) vault。下表 包含 ipa vault-* 命令的列表，并解释了它们的用途。

**注意**

在运行任何 ipa vault-* 命令前，请将密钥恢复授权 (KRA) 证书系统组件安装到 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

表 81.3. 基本 IdM vault 命令解释

| 命令 | 目的 |
| --- | --- |
| **ipa help vault** | 显示有关 IdM 库和示例密码库命令的概念信息。 |
| **ipa vault-add --help**,**ipa vault-find --help** | 在特定的 **ipa vault-*** 命令中添加 **--help** 选项会显示该命令可用的选项和详细帮助。 |
| **ipa vault-show user_vault --user idm_user** | 在将密码库作为 vault 成员访问时，您必须指定 vault 所有者。如果您没有指定 vault 所有者，IdM 会通知您没有找到密码库：<br>`[admin@server ~]$ ipa vault-show user_vault`<br>`ipa: ERROR: user_vault: vault not found` |

| 命令 | 目的 |
| --- | --- |
| **ipa vault-show shared_vault --shared** | 在访问共享密码库时，您必须指定您要访问的 vault 是共享密码库。否则，IdM 会通知您没有找到密码库：<br>[admin@server ~]$ ipa vault-show shared_vault<br>ipa: ERROR: shared_vault: vault not found |

### 81.7. 在 IDM 中安装密钥恢复授权

**按照以下流程，通过在特定的 IdM 服务器上安装密钥恢复授权(KRA)证书系统(CS)组件来在身份管理(IdM)中启用 vault。**

**先决条件**

- **您已以 root 身份登录到 IdM 服务器。**
- **IdM 证书颁发机构已安装在 IdM 服务器上。**
- **您有 目录管理器 凭证。**

**流程**

- **安装 KRA：**

```
# ipa-kra-install
```

**重要**

**您可以在隐藏的副本上安装 IdM 集群的第一个 KRA。但是，在在非隐藏的副本上安装 KRA 克隆前，安装额外的 KRA 克隆需要临时激活隐藏的副本。然后您可以再次隐藏原始隐藏的副本。**

**注意**

**要使密码库服务高可用且具有弹性，请在两个或多个 IdM 服务器上安装 KRA。维护多个 KRA 服务器可防止数据丢失。**

**其它资源**

- 请参阅 降级或提升隐藏的副本。
- 请参阅 隐藏的副本模式。

# 第 82 章 使用 IDM 用户库：存储和检索 SECRET

本章论述了如何在身份管理中使用用户库。具体来说，它描述了用户如何在 IdM 库中存储 secret，以及用户如何检索 secret。用户可以通过两个不同的 IdM 客户端进行存储和检索。

**先决条件**

- 密钥恢复授权中心 (KRA) 证书系统组件已安装在您的 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

## 82.1. 在用户密码库中存储 SECRET

按照以下流程创建带有一个或多个私有 vault 的 vault 容器，以安全地存储具有敏感信息的文件。在以下流程中使用的示例中，idm_user 用户创建标准类型的密码库。标准密码库类型确保无需 idm_user 在访问该文件时进行身份验证。idm_user 能够从用户登录的任何 IdM 客户端检索 文件。

在此过程中：

- idm_user 是想要创建密码库的用户。
- my_vault 是用于存储用户证书的库。
- vault 类型是 标准的，因此访问存档证书不要求用户提供 vault 密码。
- secret.txt 是包含用户希望在密码库中存储的证书的文件。

**先决条件**

- 您知道 idm_user 的密码。
- 您已登录到属于 IdM 客户端的主机。

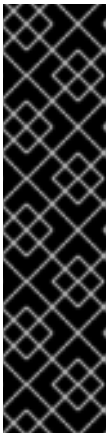
**流程**

1. 获取 idm_user 的 Kerberos 票据授予 ticket(TGT)：

```
$ kinit idm_user
```

2. 使用 ipa vault-add 命令和 --type 标准 选项来创建标准密码库：

```
$ ipa vault-add my_vault --type standard
----------------------
Added vault "my_vault"
----------------------
  Vault name: my_vault
  Type: standard
  Owner users: idm_user
  Vault user: idm_user
```



重要

确保用户的第一个用户库由同一用户创建。为用户创建第一个密码库也会创建用户的 vault 容器。创建的代理变为 vault 容器的所有者。

例如，如果其他用户（如 admin ）为 user1 创建第一个用户库，则用户的 vault 容器所有者也是 admin，并且 user1 无法访问用户密码库或创建新的用户库。

3. 使用 ipa vault-archive 命令及 --in 选项将 secret.txt 文件归档到密码库中：

```
$ ipa vault-archive my_vault --in secret.txt
---------------------------------
Archived data into vault "my_vault"
---------------------------------
```

## 82.2. 从用户密码库检索 SECRET

作为身份管理(IdM)，您可以从用户私有密码库中检索机密到您已登录的任何 IdM 客户端。

按照以下流程，以名为 idm_user 的 IdM 用户身份，将名为 my_vault 的用户私有 vault 中的 secret 检索到 idm_client.idm.example.com。

先决条件

- idm_user 是 my_vault 的所有者。
- idm_user 已在 密码库中存档了机密。
- my_vault 是一个标准密码库，这意味着 idm_user 不必输入任何密码才能访问密码库的内容。

**流程**

1. 以 idm_user 身份 SSH 到 idm_client：

```
$ ssh idm_user@idm_client.idm.example.com
```

2. 以 idm_user 身份登录：

```
$ kinit user
```

3. 使用 ipa vault-retrieve --out 命令及 --out 选项，以检索密码库的内容并将其保存到 secret_exported.txt 文件中。

```
$ ipa vault-retrieve my_vault --out secret_exported.txt
---------------------------------------
Retrieved data from vault "my_vault"
---------------------------------------
```

### 82.3. 其它资源

- 请参阅 使用 Ansible 来管理 IdM 用户 vault：存储和检索 secret。

# 第 83 章 使用 ANSIBLE 管理 IDM 用户库：存储和检索 SECRET

本章论述了如何使用 Ansible vault 模块在身份管理中管理用户密码库。具体来说，它描述了用户可以如何使用 Ansible playbook 执行以下三个连续操作：

- 在 IdM 中创建用户 vault。
- 在密码库中存储机密。
- 从密码库检索机密。

用户可以通过两个不同的 IdM 客户端进行存储和检索。

**先决条件**

- 密钥恢复授权中心 (KRA) 证书系统组件已安装在您的 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

## 83.1. 使用 ANSIBLE 在 IDM 中存在标准用户库

按照以下流程，使用 Ansible playbook 创建包含一个或多个私有 vault 的 vault 容器，以安全地存储敏感信息。在以下步骤中使用的示例中，idm_user 用户创建名为 my_vault 的标准类型库。标准密码库类型确保无需 idm_user 在访问该文件时进行身份验证。idm_user 能够从用户登录的任何 IdM 客户端检索 文件。

**先决条件**

- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包，这是您在该流程中执行步骤的主机。
- 您知道 idm_user 的密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/vault 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 创建一个清单文件，如 inventory.file ：

```
$ touch inventory.file
```

3. 打开 inventory.file，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

4. 生成 ensure-standard-vault-is-present.yml Ansible playbook 文件的副本。例如：

```
$ cp ensure-standard-vault-is-present.yml ensure-standard-vault-is-present-copy.yml
```

5. 打开 ensure-standard-vault-is-present-copy.yml 文件进行编辑。

6. 通过在 ipavault 任务部分设置以下变量来调整文件：

- 将 ipaadmin_principal 变量设置为 idm_user。
- 将 ipaadmin_password 变量设置为 idm_user 密码。
- 将 user 变量设置为 idm_user。
- 将 name 变量设置为 my_vault。
- 将 vault_type 变量设置为 standard。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
```

```
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      vault_type: standard
```

7. 保存该文件。

8. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-standard-vault-is-present-copy.yml
```

### 83.2. 使用 ANSIBLE 将 SECRET 归档到 IDM 中的标准用户库中

按照以下流程，使用 Ansible playbook 将敏感信息存储在个人 vault 中。在使用的示例中，idm_user 用户在名为 my_vault 的库中归档含有名为 password.txt 的敏感信息的文件。

**先决条件**

- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包，这是您在该流程中执行步骤的主机。
- 您知道 idm_user 的密码。
- idm_user 是所有者，或者至少是 my_vault 的成员用户。
- 您可以访问 password.txt，这是要在 my_vault 中存档的机密。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/vault 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 制作 data-archive-in-symmetric-vault.yml Ansible playbook 文件的副本，但将"symmetric"替换为 "standard"。例如：

```
$ cp data-archive-in-symmetric-vault.yml data-archive-in-standard-vault-copy.yml
```

4. 打开 data-archive-in-standard-vault-copy.yml 文件进行编辑。

5. 通过在 ipavault 任务部分设置以下变量来调整文件：

- 将 ipaadmin_principal 变量设置为 idm_user。
- 将 ipaadmin_password 变量设置为 idm_user 密码。
- 将 user 变量设置为 idm_user。
- 将 name 变量设置为 my_vault。
- 将 in 变量设置为包含敏感信息的文件的完整路径。
- 将 action 变量设置为 member。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      in: /usr/share/doc/ansible-freeipa/playbooks/vault/password.txt
      action: member
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-archive-in-standard-vault-copy.yml
```

### 83.3. 使用 ANSIBLE 从 IDM 中的标准用户库检索 SECRET

按照以下流程，使用 Ansible playbook 从用户个人 vault 检索 secret。在以下步骤中使用的示例中，idm_user 用户从名为 my_vault 的标准类型库检索包含敏感数据的文件，并检索名为 host01 的 IdM 客户端。idm_user 在访问该文件时不必进行身份验证。idm_user 可以使用 Ansible 从安装 Ansible 的任何 IdM 客户端检索 文件。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 您知道 idm_user 的密码。

- idm_user 是 my_vault 的所有者。

- idm_user 已将 secret 存储在 my_vault 中。

- Ansible 可以写入要检索该 secret 的 IdM 主机上的 目录。

- idm_user 可以从要检索 secret 的 IdM 主机上的 目录读取。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/vault 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 打开清单文件，并在一个明确定义的部分中提到您要检索该 secret 的 IdM 客户端。例如，要指示 Ansible 在 host01.idm.example.com 上检索 secret，请输入：

```
[ipahost]
host01.idm.example.com
```

3. 生成 retrive-data-symmetric-vault.yml Ansible playbook 文件的副本。将 "symmetric" 替换为 "standard"。例如：

```
$ cp retrive-data-symmetric-vault.yml retrieve-data-standard-vault.yml-copy.yml
```

4. 打开 retrieve-data-standard-vault.yml-copy.yml 文件进行编辑。

5. 通过将 hosts 变量设置为 ipahost 来调整 文件。

6. 通过在 ipavault 任务部分设置以下变量来调整文件：

- 将 ipaadmin_principal 变量设置为 idm_user。
- 将 ipaadmin_password 变量设置为 idm_user 密码。
- 将 user 变量设置为 idm_user。
- 将 name 变量设置为 my_vault。
- 将 out 变量设置为您要将 secret 导出到的文件的完整路径。
- 将 state 变量设置为 retrieve。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Tests
  hosts: ipahost
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      out: /tmp/password_exported.txt
      state: retrieved
```

7. 保存该文件。

8. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-standard-vault.yml-copy.yml
```

验证步骤

1. 以 user01 身份通过 SSH 连接到 host01：

```
$ ssh user01@host01.idm.example.com
```

2. 查看 Ansible playbook 文件中 out 变量指定的文件：

```
$ vim /tmp/password_exported.txt
```

现在，您可以看到导出的 secret。

- 有关使用 Ansible 管理 IdM vaults 和用户 secret 以及 playbook 变量的更多信息，请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-vault.md Markdown 文件，和 /usr/share/doc/ansible-freeipa/playbooks/vault/ 目录中的示例 playbook。

# 第 84 章 管理 IDM 服务 SECRET：存储和检索 SECRET

本节介绍管理员可以如何使用 ansible-freeipa vault 模块安全地将服务 secret 存储在集中式位置。示例中使用的 vault 是非对称的，这意味着要使用它，管理员需要执行以下步骤：

1. 使用 openssl 实用程序生成私钥。
2. 根据私钥生成公钥。

当管理员将服务 secret 归档到密码库时，会用公钥对其进行加密。之后，托管在域中特定计算机上的服务实例使用私钥检索该 secret。只有服务和管理员可以访问该 secret。

如果该机密泄露，管理员可以在服务 vault 中替换它，然后将它重新分发到尚未遭入侵的服务实例。

## 先决条件

- 密钥恢复授权中心 (KRA) 证书系统组件已安装在您的 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

本节包括以下步骤

1. 在非对称库中存储 IdM 服务 secret
2. 为 IdM 服务实例检索服务 secret
3. 在被破坏时更改 IdM 服务 vault secret

## 使用的术语

在流程中：

- admin 是管理服务密码的管理员。

- private-key-to-an-externally-certificate.pem 是包含服务 secret 的文件，本例中为外部签名证书的私钥。请勿将此私钥与用于从密码库检索机密的私钥混淆。
- secret_vault 是为服务创建的库。
- HTTP/webserver.idm.example.com 是正在存档其机密的服务。
- service-public.pem 是用于加密 password_vault 中存储的密码的服务公钥。
- service-private.pem 是用于解密 secret_vault 中存储的密码的服务私钥。

### 84.1. 在非对称库中存储 IDM 服务 SECRET

按照以下流程创建非对称 vault，并使用它来归档服务 secret。

**先决条件**

- 您知道 IdM 管理员密码。

**流程**

1. 以管理员身份登录：

```
$ kinit admin
```

2. 获取服务实例的公钥。例如，使用 openssl 工具：

   a. 生成 service-private.pem 私钥。

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....................................+++
e is 65537 (0x10001)
```

b. 根据私钥生成 service-public.pem 公钥。

```
$ openssl rsa -in service-private.pem -out service-public.pem -pubout
writing RSA key
```

3. 创建一个非对称密码库作为服务实例库，并提供公钥：

```
$ ipa vault-add secret_vault --service HTTP/webserver.idm.example.com --type
asymmetric --public-key-file service-public.pem
----------------------------
Added vault "secret_vault"
----------------------------
Vault name: secret_vault
Type: asymmetric
Public key: LS0tLS1C...S0tLS0tCg==
Owner users: admin
Vault service: HTTP/webserver.idm.example.com@IDM.EXAMPLE.COM
```

存档到密码库中的密码将通过 密钥进行保护。

4. 将服务 secret 归档到服务库中：

```
$ ipa vault-archive secret_vault --service HTTP/webserver.idm.example.com --in
private-key-to-an-externally-signed-certificate.pem
---------------------------------
Archived data into vault "secret_vault"
---------------------------------
```

这将使用服务实例公钥加密机密。

对需要机密的每个服务实例重复这些步骤。为每个服务实例创建一个新的非对称库。

## 84.2. 为 IDM 服务实例检索服务 SECRET

按照以下流程，使用本地存储的服务私钥，使用服务实例检索服务 vault secret。

**先决条件**

- 您可以访问拥有服务库的服务主体的 keytab，例如 HTTP/webserver.idm.example.com。
- 您已 创建了非对称密码库，并在密码库中存档了机密。
- 您可以访问用于检索服务 vault secret 的私钥。

流程

1. 以管理员身份登录：

```
$ kinit admin
```

2. 获取该服务的 Kerberos 票据：

```
# kinit HTTP/webserver.idm.example.com -k -t /etc/httpd/conf/ipa.keytab
```

3. 检索服务 vault 密码：

```
$ ipa vault-retrieve secret_vault --service HTTP/webserver.idm.example.com --private-key-file service-private.pem --out secret.txt
----------------------------------
Retrieved data from vault "secret_vault"
----------------------------------
```

### 84.3. 在被破坏时更改 IDM 服务 VAULT SECRET

按照以下流程，通过更改服务 vault secret 来隔离被入侵的服务实例。

先决条件

- 您知道 IdM 管理员密码。
- 您已 创建了用于存储服务机密的非对称密码库。
- 您已生成新 secret 并可访问它，例如：new -private-key-to-an-externally-certificate.pem

文件中。

**流程**

1. 将新 secret 归档到服务实例库中：

```
$ ipa vault-archive secret_vault --service HTTP/webserver.idm.example.com --in new-private-key-to-an-externally-signed-certificate.pem
----------------------------------
Archived data into vault "secret_vault"
----------------------------------
```

这会覆盖密码库中存储的当前机密。

2. 仅检索非代理服务实例上的新机密。详情请参阅 为 IdM 服务实例检索服务 secret。

## 84.4. 其它资源

- 请参阅 使用 Ansible 来管理 IdM 服务库：存储和检索 secret。

# 第 85 章 使用 ANSIBLE 管理 IDM 服务库：存储和检索 SECRET

本节介绍管理员可以如何使用 ansible-freeipa vault 模块安全地将服务 secret 存储在集中式位置。示例中使用的 vault 是非对称的，这意味着要使用它，管理员需要执行以下步骤：

1. 使用 openssl 实用程序生成私钥。
2. 根据私钥生成公钥。

当管理员将服务 secret 归档到密码库时，会用公钥对其进行加密。之后，托管在域中特定计算机上的服务实例使用私钥检索该 secret。只有服务和管理员可以访问该 secret。

如果该机密泄露，管理员可以在服务 vault 中替换它，然后将它重新分发到尚未遭入侵的服务实例。

**先决条件**

- 密钥恢复授权中心 (KRA) 证书系统组件已安装在您的 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

本节包括以下步骤：

- 使用 Ansible 在 IdM 中存在非对称服务库
- 使用 Ansible 将 IdM 服务 secret 存储在非对称库中
- 使用 Ansible 为 IdM 服务检索服务 secret
- 在使用 Ansible 泄露时更改 IdM 服务 vault secret

在流程中：

- admin 是管理服务密码的管理员。
- private-key-to-an-externally-certificate.pem 是包含服务 secret 的文件，本例中为外部签名证书的私钥。请勿将此私钥与用于从密码库检索机密的私钥混淆。
- secret_vault 是为存储服务 secret 而创建的库。
- HTTP/webserver1.idm.example.com 是密码库的所有者服务。
- HTTP/webserver2.idm.example.com 和 HTTP/webserver3.idm.example.com 是 vault 成员服务。
- service-public.pem 是用于加密 password_vault 中存储的密码的服务公钥。
- service-private.pem 是用于解密 secret_vault 中存储的密码的服务私钥。

## 85.1. 使用 ANSIBLE 在 IDM 中存在非对称服务库

按照以下流程，使用 Ansible playbook 创建包含一个或多个私有 vault 容器的服务 vault 容器，以安全地存储敏感信息。在以下流程中使用的示例中，管理员创建名为 secret_vault 的非对称库。这样可确保 vault 成员必须使用私钥进行身份验证，以检索密码库中的 secret。vault 成员能够从任何 IdM 客户端检索 文件。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/vault 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 获取服务实例的公钥。例如，使用 openssl 工具：

   a. 生成 service-private.pem 私钥。

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....................................+++
e is 65537 (0x10001)
```

   b. 根据私钥生成 service-public.pem 公钥。

```
$ openssl rsa -in service-private.pem -out service-public.pem -pubout
writing RSA key
```

3. 可选：如果不存在，创建一个清单文件（如 inventory.file）。

```
$ touch inventory.file
```

4. 打开清单文件，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

5. 生成 ensure-asymmetric-vault-is-present.yml Ansible playbook 文件的副本。例如：

```
$ cp ensure-asymmetric-vault-is-present.yml ensure-asymmetric-service-vault-is-present-copy.yml
```

6. 打开 ensure-asymmetric-vault-is-present-copy.yml 文件进行编辑。

7. 添加一个任务，该任务将 service-public.pem 公钥从 Ansible 控制器复制到 server.idm.example.com 服务器。

8. 通过在 ipavault 任务部分设置以下变量来修改文件的其余部分：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 使用 name 变量定义 vault 的名称，如 secret_vault。
- 将 vault_type 变量设置为非 对称。
- 将 service 变量设置为拥有密码库的服务主体，如 HTTP/webserver1.idm.example.com。
- 将 public_key_file 设置为您的公钥的位置。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Copy public key to ipaserver.
    copy:
```

```
    src: /path/to/service-public.pem
    dest: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
    mode: 0600
- name: Add data to vault, from a LOCAL file.
  ipavault:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: secret_vault
    vault_type: asymmetric
    service: HTTP/webserver1.idm.example.com
    public_key_file: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
```

9. 保存该文件。

10. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-asymmetric-service-vault-is-present-copy.yml
```

### 85.2. 使用 ANSIBLE 将成员服务添加到非对称库

按照以下流程，使用 Ansible playbook 将成员服务添加到服务 vault 中，以便它们都可以检索 vault 中存储的 secret。在以下流程中使用的示例中，IdM 管理员将 HTTP/webserver2.idm.example.com 和 HTTP/webserver3.idm.example.com 服务主体添加到由 HTTP/webserver1.idm.example.com 所有的 secret_vault vault 中。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
-

目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 您知道 IdM 管理员密码。
- 您已创建了非对称密码库用于存储服务机密。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/vault 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 可选：如果不存在，创建一个清单文件（如 inventory.file）。

```
$ touch inventory.file
```

3. 打开清单文件，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

4. 生成 data-archive-in-asymmetric-vault.yml Ansible playbook 文件的副本。例如：

```
$ cp data-archive-in-asymmetric-vault.yml add-services-to-an-asymmetric-vault.yml
```

5. 打开 data-archive-in-asymmetric-vault-copy.yml 文件进行编辑。

6. 通过在 ipavault 任务部分设置以下变量来修改该文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员密码。
   - 将 name 变量设置为 vault 的名称，如 secret_vault。

- 将 service 变量设置为密码库的服务所有者，如 HTTP/webserver1.idm.example.com。

- 定义您要使用 services 变量访问 vault 机密的服务。

- 将 action 变量设置为 member。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      services:
      - HTTP/webserver2.idm.example.com
      - HTTP/webserver3.idm.example.com
      action: member
```

7. 保存该文件。

8. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file add-services-to-an-asymmetric-vault.yml
```

### 85.3. 使用 ANSIBLE 将 IDM 服务 SECRET 存储在非对称库中

按照以下流程，使用 Ansible playbook 将 secret 存储在服务 vault 中，以便稍后可被服务检索。在以下流程中使用的示例中，管理员将带有 secret 的 PEM 文件存储在名为 secret_vault 的非对称库中。这样可确保服务必须使用私钥进行身份验证，以便从vault 中检索 secret 。vault 成员能够从任何 IdM 客户端检索 文件。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。
- 您已创建了非对称密码库用于存储服务机密。
- secret 存储在 Ansible 控制器上，例如 /usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-certificate.pem 文件中。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/vault 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 可选：如果不存在，创建一个清单文件（如 inventory.file）。

```
$ touch inventory.file
```

3. 打开清单文件，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

4. 生成 data-archive-in-asymmetric-vault.yml Ansible playbook 文件的副本。例如：

```
$ cp data-archive-in-asymmetric-vault.yml data-archive-in-asymmetric-vault-copy.yml
```

5. 打开 data-archive-in-asymmetric-vault-copy.yml 文件进行编辑。

6. 通过在 ipavault 任务部分设置以下变量来修改该文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 name 变量设置为 vault 的名称，如 secret_vault。
- 将 service 变量设置为密码库的服务所有者，如 HTTP/webserver1.idm.example.com。
- 将 in 变量设置为 "{{ lookup('file', 'private-key-to-an-externally-certificate.pem')| b64encode }}"。这可确保 Ansible 使用私钥从 Ansible 控制器上的工作目录检索文件，而不是从 IdM 服务器检索。
- 将 action 变量设置为 member。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
```

```
    name: secret_vault
    service: HTTP/webserver1.idm.example.com
    in: "{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem') |
b64encode }}"
    action: member
```

7. 保存该文件。

8. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-
archive-in-asymmetric-vault-copy.yml
```

## 85.4. 使用 ANSIBLE 为 IDM 服务检索服务 SECRET

按照以下流程，使用 Ansible playbook 代表服务从服务 vault 中检索 secret。在以下流程中使用的示例中，运行 playbook 从名为 secret_vault 的非对称库检索带有 secret 的 PEM 文件，并将它存储在 Ansible 清单文件中列出的所有主机上的指定位置，存为 ipaservers。

服务使用 keytabs 验证 IdM，并使用私钥与密码库进行身份验证。您可以代表服务从安装 ansible-freeipa 的任何 IdM 客户端检索 文件。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、

服务器或副本的一部分。

- 您知道 IdM 管理员密码。
- 您已创建了非对称密码库用于存储服务机密。
- 您已在密码库中存档了机密。
- 您已将用于检索服务 vault secret 的私钥存储在 Ansible 控制器上的 private_key_file 变量指定的位置。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/vault 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 可选：如果不存在，创建一个清单文件（如 inventory.file）。

```
$ touch inventory.file
```

3. 打开清单文件并定义以下主机：

   - 在 [ipaserver] 部分中定义您的 IdM 服务器。
   - 在 [webservers] 部分中定义要检索机密的主机。例如，要指示 Ansible 获取到 webserver1.idm.example.com、webserver2.idm.example.com 和 webserver3.idm.example.com 的 secret，请输入：

```
[ipaserver]
server.idm.example.com

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
webserver3.idm.example.com
```

4. 生成 retrieve-data-asymmetric-vault.yml Ansible playbook 文件的副本。例如：

```
$ cp retrieve-data-asymmetric-vault.yml retrieve-data-asymmetric-vault-copy.yml
```

5. 打开 retrieve-data-asymmetric-vault-copy.yml 文件进行编辑。

6. 通过在 ipavault 任务部分设置以下变量来修改该文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 name 变量设置为 vault 的名称，如 secret_vault。
- 将 service 变量设置为密码库的服务所有者，如 HTTP/webserver1.idm.example.com。
- 将 private_key_file 变量设置为用于检索服务 vault secret 的私钥的位置。
- 将 out 变量设置为 IdM 服务器上您要检索 private-key-to-an-externally-certificate.pem 机密的位置，如当前工作目录。
- 将 action 变量设置为 member。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Retrieve data from the service vault
    ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
```

```
    service: HTTP/webserver1.idm.example.com
    vault_type: asymmetric
    private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
    out: private-key-to-an-externally-signed-certificate.pem
    state: retrieved
```

7. 在 playbook 中添加一个部分，它将从 IdM 服务器检索数据文件到 Ansible 控制器：

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false
  tasks:
[...]
  - name: Retrieve data file
    fetch:
      src: private-key-to-an-externally-signed-certificate.pem
      dest: ./
      flat: yes
      mode: 0600
```

8. 在 playbook 中添加一个部分，它将检索到的 private-key-to-an-externally-certificate.pem 文件从 上的 Ansible 控制器传输到清单文件的 webservers 部分：

```
---
- name: Send data file to webservers
  become: no
  gather_facts: no
  hosts: webservers
  tasks:
  - name: Send data to webservers
    copy:
      src: private-key-to-an-externally-signed-certificate.pem
      dest: /etc/pki/tls/private/httpd.key
      mode: 0444
```

9. 保存该文件。

10. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-
data-asymmetric-vault-copy.yml
```

## 85.5. 在使用 ANSIBLE 泄露时更改 IDM 服务 VAULT SECRET

当服务实例被破坏时，请按照此流程重复使用 Ansible playbook 来更改存储在服务 vault 中的 secret。以下示例中的情景假定 on webserver3.idm.example.com 检索到的机密已被破坏，而不是存储该机密的非对称库的密钥。在示例中，管理员重复利用在非对称库中存储一个 secret 时，以及从非对称库中获取一个 secret 导入到 IdM 主机 时使用的 Ansible playbook。在流程开始时，IdM 管理员使用非对称密码库中的新 secret 存储一个新的 PEM 文件，调整清单文件，而不检索到被入侵的 Web 服务器 webserver3.idm.example.com，然后重新运行这两个程序。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。
- 您已创建了非对称密码库用于存储服务机密。
- 您已为在 IdM 主机上运行的 web 服务生成了新的 httpd 密钥，以替换泄露的旧密钥。
- 新的 httpd 密钥存储在 Ansible 控制器上，例如 /usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-certificate.pem 文件中。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/vault 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 打开清单文件，并确保正确定义了以下主机：

- [ipaserver] 部分中的 IdM 服务器。
- 要检索 [webservers] 部分中的机密的主机。例如，要指示 Ansible 检索到 webserver1.idm.example.com 和 webserver2.idm.example.com 的 secret，请输入：

```
[ipaserver]
server.idm.example.com

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
```

重要

确保列表不包含当前 example webserver3.idm.example.com 中被入侵的 web 服务器。

3. 打开 data-archive-in-asymmetric-vault-copy.yml 文件进行编辑。

4. 通过在 ipavault 任务部分设置以下变量来修改该文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 name 变量设置为 vault 的名称，如 secret_vault。
- 将 service 变量设置为密码库的服务所有者，如 HTTP/webserver.idm.example.com。
-

将 中的变量 设置为 "{{ lookup('file', 'new-private-key-to-an-externally-certificate.pem')| b64encode }}"。这可确保 Ansible 使用私钥从 Ansible 控制器上的工作目录检索文件，而不是从 IdM 服务器检索。

- 将 action 变量设置为 member。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver.idm.example.com
      in: "{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem') |
b64encode }}"
      action: member
```

5. 保存该文件。

6. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-
archive-in-asymmetric-vault-copy.yml
```

7. 打开 retrieve-data-asymmetric-vault-copy.yml 文件进行编辑。

8. 通过在 ipavault 任务部分设置以下变量来修改该文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 name 变量设置为 vault 的名称，如 secret_vault。
-

将 service 变量设置为密码库的服务所有者，如 HTTP/webserver1.idm.example.com。

- 将 private_key_file 变量设置为用于检索服务 vault secret 的私钥的位置。
- 将 变量 设置为 IdM 服务器上您要检索 new-private-key-to-an-externally-certificate.pem 机密的位置，如当前工作目录。
- 将 action 变量设置为 member。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Retrieve data from the service vault
    ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      vault_type: asymmetric
      private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
      out: new-private-key-to-an-externally-signed-certificate.pem
      state: retrieved
```

9. 在 playbook 中添加一个部分，它将从 IdM 服务器检索数据文件到 Ansible 控制器：

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: true
  gather_facts: false
  tasks:
[...]
  - name: Retrieve data file
    fetch:
      src: new-private-key-to-an-externally-signed-certificate.pem
      dest: ./
      flat: yes
      mode: 0600
```

10. 在 playbook 中添加一个部分，它将检索到的 new-private-key-to-an-externally-certificate.pem 文件从 上的 Ansible 控制器传输到清单文件的 webservers 部分：

```
---
- name: Send data file to webservers
  become: true
  gather_facts: no
  hosts: webservers
  tasks:
  - name: Send data to webservers
    copy:
      src: new-private-key-to-an-externally-signed-certificate.pem
      dest: /etc/pki/tls/private/httpd.key
      mode: 0444
```

11. 保存该文件。

12. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

## 85.6. 其它资源

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-vault.md Markdown 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/vault/ 目录中的 playbook 示例。

# 第 86 章 使用 ANSIBLE 在 IDM 中确保存在或不存在服务

使用 Ansible service 模块时，管理员可以确保 IdM 中存在或不存在不原生 IdM 的特定服务。例如，您可以使用 service 模块：

- 检查 IdM 客户端中是否存在手动安装的服务，并在缺少该服务时自动安装该服务。详情请查看：
  - 确保 IdM 客户端的 IdM 中存在 HTTP 服务。
  - 使用单个 Ansible 任务，确保多个服务在 IdM 客户端上的 IdM 中存在。
  - 确保非 IdM 客户端的 IdM 中存在 HTTP 服务。
  - 确保在没有 DNS 的 IdM 客户端中存在 HTTP 服务。
- 检查在 IdM 中注册的服务是否已附加证书，并在缺少该证书时自动安装该证书。详情请查看：
  - 确保 IdM 服务条目中存在外部签名的证书。
- 允许 IdM 用户和主机检索并创建服务 keytab。详情请查看：
  - 允许 IdM 用户、组、主机或主机组创建服务的 keytab。
  - 允许 IdM 用户、组、主机或主机组检索服务的 keytab。
- 允许 IdM 用户和组向服务添加 Kerberos 别名。详情请查看：
  - 确保服务的 Kerberos 主体别名存在。

- 检查 IdM 客户端中是否不存在服务，并在该服务存在时自动删除该服务。详情请查看：
  - 确保 IdM 客户端的 IdM 中缺少 HTTP 服务。

## 86.1. 使用 ANSIBLE PLAYBOOK 在 IDM 中存在 HTTP 服务

按照以下流程，使用 Ansible playbook 确保 HTTP 服务器在 IdM 中存在。

### 先决条件

- 托管 HTTP 服务的系统是一个 IdM 客户端。
- 您有 IdM 管理员密码。

### 流程

1. 创建一个清单文件，如 inventory.file ：

```
$ touch inventory.file
```

2. 打开 inventory.file，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present.yml Ansible playbook 文件的副本。例如：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present.yml
/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-copy.yml
```

4. 打开 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-copy.yml Ansible playbook 文件进行编辑：

```
---
```

```
- name: Playbook to manage IPA service.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service is present
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
```

5. 调整文件：

   - 更改 ipaadmin_password 变量定义的 IdM 管理员密码。
   - 更改运行 HTTP 服务的 IdM 客户端的名称，如 ipaservice 任务 的名称 变量所定义。

6. 保存并退出文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-is-present-copy.yml
```

验证步骤

1. 以 IdM 管理员身份登录 IdM Web UI。

2. 导航到 Identity → Services。

如果 Services 列表中列出了 HTTP/client.idm.example.com@IDM.EXAMPLE.COM，Ansible playbook 已成功添加到 IdM。

其它资源

- 为了保护 HTTP 服务器和浏览器客户端之间的通信，请参阅 向 Apache HTTP 服务器 添加

TLS 加密。

- 要为 HTTP 服务请求证书，请参阅 使用 certmonger 来为服务获取 IdM 证书 中的描述的流程。

## 86.2. 使用单个 ANSIBLE 任务，确保多个服务在 IDM 客户端上的 IDM 中存在

您可以使用 ansible-freeipa ipaservice 模块，使用单个 Ansible 任务添加、修改和删除多个身份管理(IdM)服务。为此，请使用 ipaservice 模块的 services 选项。

使用 services 选项，您还可以指定多个仅应用到某个特定服务的服务变量。通过 name 变量定义此服务，这是 services 选项的唯一强制变量。

完成此流程，以使用单个任务确保 HTTP/client01.idm.example.com@IDM.EXAMPLE.COM 和 ftp/client02.idm.example.com@IDM.EXAMPLE.COM 服务在 IdM 中存在。

**先决条件**

- 在控制节点上：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已安装 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您正在使用 RHEL 8.9 及更新版本。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。

**流程**

1. 使用以下内容创建您的 Ansible playbook 文件 add-http-and-ftp-services.yml ：

```
---
- name: Playbook to add multiple services in a single task
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Add HTTP and ftp services
    ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      services:
      - name:  HTTP/client01.idm.example.com@IDM.EXAMPLE.COM
      - name: ftp/client02.idm.example.com@IDM.EXAMPLE.COM
```

2. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-http-and-ftp-services.yml
```

其它资源

- ansible-freeipa 上游文档中的 service 模块

### 86.3. 使用 ANSIBLE PLAYBOOK，确保在 IDM 中存在于非IDM 客户端中的 HTTP 服务

按照以下流程，使用 Ansible playbook 确保 IdM 中的 HTTP 服务器在不是 IdM 客户端的主机上存在。通过将 HTTP 服务器添加到 IdM 中，您还会将主机添加到 IdM。

先决条件

- 您已在主机上 安装了 HTTP 服务。
- 设置 HTTP 的主机不是 IdM 客户端。否则，请按照 使用 Ansible playbook 在 IdM 中保证 HTTP 服务存在中的步骤进行操作。
- 您有 IdM 管理员密码。

- 主机的 DNS A 记录 - 或 AAAA 记录（如果使用 IPv6）

**流程**

1. 创建一个清单文件，如 inventory.file ：

```
$ touch inventory.file
```

2. 打开 inventory.file，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check.yml Ansible playbook 文件的副本。例如：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml
```

4. 打开复制的文件 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml 以进行编辑。在 ipaservice 任务中找到 ipaadmin_password 和 name 变量：

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service is present
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/www2.example.com
      skip_host_check: yes
```

5. 调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。

- 将 name 变量设置为运行 HTTP 服务的主机的名称。

6. 保存并退出文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-is-present-without-host-check-copy.yml
```

验证步骤

1. 以 IdM 管理员身份登录 IdM Web UI。

2. 导航到 Identity → Services。

现在，您可以看到 Services 列表中列出的 HTTP/client.idm.example.com@IDM.EXAMPLE.COM。

其它资源

- 要保护通信，请参阅 向 Apache HTTP 服务器 添加 TLS 加密。

## 86.4. 使用 ANSIBLE PLAYBOOK 确保在没有 DNS 的 IDM 客户端上存在 HTTP 服务

按照以下流程，使用 Ansible playbook 确保运行在没有 DNS 条目的 IdM 客户端上的 HTTP 服务器存在。场景表示，如果使用 IPv6 而不是 IPv4，IdM 主机没有可用的 DNS A 条目 - 或没有 DNS AAAA 条目。

先决条件

- 托管 HTTP 服务的系统已在 IdM 中注册。

- 主机的 DNS A 或 DNS AAAA 记录可能不存在。否则，如果主机的 DNS 记录存在，请按照

以下步骤 确保使用 Ansible playbook 在 IdM 中存在 HTTP 服务。

- 您有 IdM 管理员密码。

**流程**

1. 创建一个清单文件，如 inventory.file ：

```
$ touch inventory.file
```

2. 打开 inventory.file，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force.yml Ansible playbook 文件的副本。例如：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml
```

4. 打开复制的文件 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml，以进行编辑。在 ipaservice 任务中找到 ipaadmin_password 和 name 变量：

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service is present
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/ihavenodns.info
      force: yes
```

5. 调整文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员密码。
   - 将 name 变量设置为运行 HTTP 服务的主机的名称。

6. 保存并退出文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-is-present-with-host-force-copy.yml
```

验证步骤

1. 以 IdM 管理员身份登录 IdM Web UI。
2. 导航到 Identity → Services。

现在，您可以看到 Services 列表中列出的 HTTP/client.idm.example.com@IDM.EXAMPLE.COM。

其它资源

- 要保护通信，请参阅 向 Apache HTTP 服务器 添加 TLS 加密。

## 86.5. 使用 ANSIBLE PLAYBOOK 确保 IDM 服务条目中存在外部签名的证书

按照以下流程，使用 ansible-freeipa service 模块确保外部证书颁发机构(CA)发布的证书附加到 HTTP 服务的 IdM 条目。如果您的 IdM CA 使用自签名证书，则由外部 CA 而不是 IdM CA 签名的 HTTP 服务证书特别有用。

先决条件

- 您已在主机上 安装了 HTTP 服务。
- 您已 将 HTTP 服务注册到 IdM。
- 您有 IdM 管理员密码。
- 您有一个外部签名的证书，其 Subject 对应于 HTTP 服务的主体。

**流程**

1. 创建一个清单文件，如 inventory.file ：

```
$ touch inventory.file
```

2. 打开 inventory.file，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present.yml 文件的副本，例如：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present-copy.yml
```

4. 可选： 如果证书采用 Privacy Enhanced Mail(PEM)格式，请将证书转换为可辨识的编码规则(DER)格式，以便通过命令行界面(CLI)更轻松地处理：

```
$ openssl x509 -outform der -in cert1.pem -out cert1.der
```

5. 使用 base64 命令将 DER 文件解码为标准输出。使用 -w0 选项禁用换行：

```
$ base64 cert1.der -w0
MIIC/zCCAeegAwIBAgIUV74O+4kXeg21o4vxfRRtyJm...
```

6. 将证书从标准输出复制到剪贴板。

7. 打开 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present-copy.yml 文件进行编辑并查看其内容：

```
---
- name: Service certificate present.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service certificate is present
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
      certificate: |
        - MIICBjCCAW8CFHnm32VcXaUDGfEGdDL/...
      [...]
      action: member
      state: present
```

8. 调整文件：

- 将使用 证书 变量定义的证书替换为您从 CLI 复制的证书。请注意，如果您使用带有所示"|"管道字符的 certificate: 变量，您可以输入证书 THIS WAY，而不是让它在一个行中输入。这样可以更轻松地读取证书。
- 更改由 ipaadmin_password 变量定义的 IdM 管理员密码。
- 更改运行 HTTP 服务的 IdM 客户端的名称，由 name 变量定义。
- 更改任何其他相关变量。

9. 保存并退出文件。

10. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-member-certificate-present-copy.yml
```

验证步骤

1. 以 IdM 管理员身份登录 IdM Web UI。
2. 导航到 Identity → Services。
3. 使用新添加的证书，单击服务的名称，如 HTTP/client.idm.example.com。

在右侧的 Service Certificate 部分中，您现在可以看到新添加的证书。

## 86.6. 使用 ANSIBLE PLAYBOOK 来允许 IDM 用户、组、主机或主机组创建服务的 KEYTAB

keytab 是一个包含 Kerberos 主体和加密密钥对的文件。keytab 文件通常用于允许脚本使用 Kerberos 自动进行身份验证，无需人工交互或访问存储在纯文本文件中的密码。然后，脚本可以使用获取的凭据来访问存储在远程系统上的文件。

作为身份管理(IdM)管理员，您可以允许其他用户为 IdM 中运行的服务检索甚至创建 keytab。通过允许特定用户和用户组创建 keytab，您可以将服务管理委派给他们，而无需共享 IdM 管理员密码。此委派提供了更加精细的系统管理。

按照以下流程，允许特定的 IdM 用户、用户组、主机和主机组为运行在 IdM 客户端上的 HTTP 服务创建 keytab。具体来说，它描述了如何允许 user01 IdM 用户为名为 client.idm.example.com 的 IdM 客户端上运行的 HTTP 服务创建 keytab。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本

您使用 Ansible 版本 2.14 或更高版本。

- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您已 将 HTTP 服务注册到 IdM。
- 托管 HTTP 服务的系统是一个 IdM 客户端。
- IdM 中已存在您要允许创建 keytab 的 IdM 用户和用户组。
- IdM 中已存在您要允许创建 keytab 的 IdM 主机和主机组。

**流程**

1. 创建一个清单文件，如 inventory.file ：

```
$ touch inventory.file
```

2. 打开 inventory.file，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present.yml Ansible playbook 文件的副本。例如：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-member-
allow_create_keytab-present.yml /usr/share/doc/ansible-
freeipa/playbooks/service/service-member-allow_create_keytab-present-copy.yml
```

4. 打开 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present-copy.yml Ansible playbook 文件进行编辑。

5. 通过更改以下内容来调整文件：

- 由 ipaadmin_password 变量指定的 IdM 管理员密码。
- 运行 HTTP 服务的 IdM 客户端的名称。在当前示例中，它是 HTTP/client.idm.example.com
- allow_create_keytab_user: 部分中列出的 IdM 用户名称。在当前示例中，是 user01。
- allow_create_keytab_group: 部分中列出的 IdM 用户组名称。
- allow_create_keytab_host: 部分中列出的 IdM 主机名称。
- allow_create_keytab_hostgroup: 部分中所列的 IdM 主机组名称。
- 由 tasks 部分中 name 变量指定的任务名称。

在适应当前示例后，复制的文件类似如下：

```
---
- name: Service member allow_create_keytab present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Service HTTP/client.idm.example.com members allow_create_keytab present
```

```
for user01
   ipaservice:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: HTTP/client.idm.example.com
    allow_create_keytab_user:
    - user01
    action: member
```

6. 保存该文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-member-allow_create_keytab-present-copy.yml
```

验证步骤

1. 以 IdM 用户身份 SSH 到 IdM 服务器，该用户具有为特定 HTTP 服务创建 keytab 的权限：

```
$ ssh user01@server.idm.example.com
Password:
```

2. 使用 ipa-getkeytab 命令为 HTTP 服务生成新 keytab:

```
$ ipa-getkeytab -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

s 选项指定用于 生成 keytab 的密钥分发中心(KDC)服务器。

p 选项指定 您要创建的 keytab 主体。

k 选项指定 将新密钥附加到的 keytab 文件。如果文件不存在，则会创建此文件。

如果命令不产生错误，您以 user01 身份成功创建了 HTTP/client.idm.example.com 的 keytab。

## 86.7. 使用 ANSIBLE PLAYBOOK 来允许 IDM 用户、组、主机或主机组检索服务的 KEYTAB

keytab 是一个包含 Kerberos 主体和加密密钥对的文件。keytab 文件通常用于允许脚本使用 Kerberos 自动进行身份验证，无需人工交互或访问存储在纯文本文件中的密码。然后，脚本可以使用获取的凭据来访问存储在远程系统上的文件。

作为 IdM 管理员，您可以允许其他用户为 IdM 中运行的服务检索甚至创建 keytab。

按照以下流程，允许特定的 IdM 用户、用户组、主机和主机组检索运行在 IdM 客户端上的 HTTP 服务的 keytab。具体来说，它描述了如何允许 user01 IdM 用户检索 client.idm.example.com 上运行的 HTTP 服务的 keytab。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您已 将 HTTP 服务注册到 IdM。
- IdM 中已存在您要允许检索 keytab 的 IdM 用户和用户组。

- IdM 中已存在您要允许检索 keytab 的 IdM 主机和主机组。

流程

1. 创建一个清单文件，如 inventory.file ：

```
$ touch inventory.file
```

2. 打开 inventory.file，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present.yml Ansible playbook 文件的副本。例如：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present-copy.yml
```

4. 打开复制的文件 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present-copy.yml 以进行编辑：

5. 调整文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员密码。
   - 将 ipaservice 任务的 name 变量设置为 HTTP 服务的主体。在当前示例中，它是 HTTP/client.idm.example.com
   - 在 allow_retrieve_keytab_group: 部分中指定 IdM 用户的名称。在当前示例中，是 user01。
   - 在 allow_retrieve_keytab_group: 部分中指定 IdM 用户组的名称。

- 在 allow_retrieve_keytab_group: 部分中指定 IdM 主机的名称。
- 在 allow_retrieve_keytab_group: 部分中指定 IdM 主机组的名称。
- 使用 tasks 部分中的 name 变量指定 任务的名称。

在适应当前示例后，复制的文件类似如下：

```
---
- name: Service member allow_retrieve_keytab present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Service HTTP/client.idm.example.com members allow_retrieve_keytab
present for user01
    ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
      allow_retrieve_keytab_user:
      - user01
      action: member
```

6. 保存该文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-member-allow_retrieve_keytab-present-copy.yml
```

验证步骤

1. 以 IdM 用户身份 SSH 到 IdM 服务器，并具有权限检索 HTTP 服务的 keytab:

```
$ ssh user01@server.idm.example.com
Password:
```

2. 使用 ipa-getkeytab 命令和 -r 选项来检索 keytab：

```
$ ipa-getkeytab -r -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

s 选项指定 您要从中检索 keytab 的密钥分发中心(KDC)服务器。

p 选项指定 您要检索的 keytab 主体。

k 选项指定 您要将检索到的密钥附加到的 keytab 文件。如果文件不存在，则会创建此文件。

如果命令不产生错误，您以 user01 身份成功检索了 HTTP/client.idm.example.com 的 keytab。

## 86.8. 使用 ANSIBLE PLAYBOOK 确保存在服务的 KERBEROS 主体别名

在某些情况下，IdM 管理员可启用 IdM 用户、主机或服务使用 Kerberos 主体别名进行身份验证。这些情况包括：

- 用户名已更改，但该用户应该能够使用先前和新用户名登录系统。
- 即使 IdM Kerberos 域与电子邮件域不同，用户也需要使用电子邮件地址登录。

按照以下流程，为运行 client.idm.example.com 上的 HTTP 服务创建 HTTP/mycompany.idm.example.com 主体别名。

### 先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。

- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您已 设置 HTTP 服务
- 您已 将 HTTP 服务注册到 IdM。
- 设置 HTTP 的主机是一个 IdM 客户端。

**流程**

1. 创建一个清单文件，如 inventory.file ：

```
$ touch inventory.file
```

2. 打开 inventory.file，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present.yml Ansible playbook 文件的副本。例如：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-copy.yml
```

4. 打开 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-copy.yml Ansible playbook 文件进行编辑。

5. 通过更改以下内容来调整文件：

- 由 ipaadmin_password 变量指定的 IdM 管理员密码。
- 通过 name 变量指定的服务名称。这是服务的规范主体名称。在当前示例中，它是 HTTP/client.idm.example.com。
- 由主体变量指定的 Kerberos 主体 别名。这是您要添加到 name 变量定义的服务的别名。在当前示例中，它是 host/mycompany.idm.example.com。
- 由 tasks 部分中 name 变量指定的任务名称。

在适应当前示例后，复制的文件类似如下：

```
---
- name: Service member principal present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Service HTTP/client.idm.example.com member principals
host/mycompany.idm.exmaple.com present
    ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
      principal:
        - host/mycompany.idm.example.com
      action: member
```

6. 保存该文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-member-principal-present-copy.yml
```

如果运行 playbook 会导致 0 个无法访问和 0 个失败的任务，您已成功为 HTTP/client.idm.example.com 服务创建了 host/mycompany.idm.example.com Kerberos 主体。

**其它资源**

- 请参阅 为用户、主机和服务管理 Kerberos 主体别名。

## 86.9. 使用 ANSIBLE PLAYBOOK 确保 IDM 中缺少 HTTP 服务

按照以下流程从 IdM 取消服务的注册。更具体地说，它描述了如何使用 Ansible playbook 来确保 IdM 中缺少名为 HTTP/client.idm.example.com 的 HTTP 服务器。

**先决条件**

- 您有 IdM 管理员密码。

**流程**

1. 创建一个清单文件，如 inventory.file ：

```
$ touch inventory.file
```

2. 打开 inventory.file，并在 [ipaserver] 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent.yml Ansible playbook 文件的副本。例如：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent.yml
/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent-copy.yml
```

4.

打开 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent-copy.yml Ansible playbook 文件进行编辑。

5. 通过更改以下内容来调整文件：

- 由 ipaadmin_password 变量定义的 IdM 管理员密码。
- HTTP 服务的 Kerberos 主体，由 ipaservice 任务 的名称 变量定义。

在适应当前示例后，复制的文件类似如下：

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service is absent
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
      state: absent
```

6. 保存并退出文件。

7. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-is-absent-copy.yml
```

验证步骤

1. 以 IdM 管理员身份登录 IdM Web UI。

2. 导航到 Identity → Services。

如果您无法在 Services 列表中看到 HTTP/client.idm.example.com@IDM.EXAMPLE.COM 服务，则已成功确保了在 IdM 中缺少 HTTP/client.idm.example.com@IDM.EXAMPLE.COM 服务。

## 86.10. 其它资源

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-service.md Markdown 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/config 目录中的 playbook 示例。

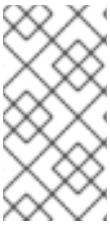
# 第 87 章 启用 AD 用户管理 IDM

## 87.1. AD 用户的 ID 覆盖

在 Red Hat Enterprise Linux(RHEL)7 中，外部组成员资格允许 Active Directory(AD)用户和组在 System Security Services Daemon(SSSD)的帮助下访问 POSIX 环境中的身份管理(IdM)资源。

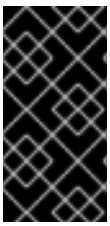
IdM LDAP 服务器具有自己的机制来授予访问控制。RHEL 8 引进了一个更新，它许作为 IdM 组成员为 AD 用户添加 ID 用户覆盖。ID 覆盖是一种记录，描述了特定的活动目录用户或组属性在特定 ID 视图（本例中为 Default Trust View ）中应该是什么样子。更新后，IdM LDAP 服务器可以为 AD 用户应用 IdM 组的访问控制规则。

AD 用户现在可以使用 IdM UI 的自助服务功能，例如上传其 SSH 密钥或更改其个人数据。AD 管理员可以在没有两个不同的帐户和密码的情况下完全管理 IdM。



注意

目前，IdM 中选定的功能可能仍对 AD 用户不可用。例如，将 IdM 用户的密码设置为 IdM admins 组中的 AD 用户可能会失败。



重要

不要将 AD 用户的 ID 覆盖用于 IdM 中的 sudo 规则。AD 用户的 ID 覆盖只代表 AD 用户的 POSIX 属性，而不是 AD 用户本身。

其它资源

- 为活动目录用户使用 ID 视图

## 87.2. 使用 ID 覆盖来启用 AD 用户管理 IDM

按照以下流程，为 AD 用户创建和使用 ID 覆盖，以给该用户授予与 IdM 用户相同的权利。在此过程中，可在配置为信任控制器或信任代理的 IdM 服务器中工作。

先决条件

- 在身份管理(IdM)服务器上启用了 idm:DL1 流，您切换到通过这个流提供的 RPM：

```
# yum module enable idm:DL1
# yum distro-sync
```

- idm:DL1/adtrust 配置集安装在 IdM 服务器上。

```
# yum module install idm:DL1/adtrust
```

该配置集包含安装与 Active Directory (AD)具有信任协议的 IdM 服务器所需的所有软件包。

- 设置了一个有效的 IdM 环境。详情请参阅 安装身份管理。
- 您的 IdM 环境与 AD 之间设置了有效信任。

**流程**

1. 作为 IdM 管理员，在 Default Trust View 中为 AD 用户创建一个 ID 覆盖。例如，要为用户 ad_user@ad.example.com 创建 ID 覆盖：

```
# kinit admin
# ipa idoverrideuser-add 'default trust view' ad_user@ad.example.com
```

2. 添加 Default Trust View 中的 ID 覆盖作为 IdM 组的成员。这必须是非 POSIX 组，因为它与 Active Directory 交互。

如果问题中的组是 IdM 角色的成员，则 ID 覆盖所代表的 AD 用户在使用 IdM API 时获得角色授予的所有权限，包括命令行界面和 IdM Web UI。

例如，要将 ad_user@ad.example.com 用户的 ID 覆盖添加到 IdM admins 组中：

```
# ipa group-add-member admins --idoverrideusers=ad_user@ad.example.com
```

3. 或者，您可以在角色中添加 ID 覆盖，如 User Administrator 角色：

```
# ipa role-add-member 'User Administrator' --
idoverrideusers=ad_user@ad.example.com
```

**其它资源**

- 为活动目录用户使用 ID 视图

### 87.3. 使用 ANSIBLE 启用 AD 用户管理 IDM

按照以下流程，使用 Ansible playbook 确保户 ID 覆盖在身份管理(IdM)组中存在。用户 ID 覆盖是您在使用 AD 建立信任视图中创建的 Active Directory (AD)用户覆盖。因此，运行 playbook （如 AD 用户）能够完全管理 IdM，而无需两个不同的帐户和密码。

先决条件

- 您需要知道 IdM admin 密码。
- 已使用 AD 安装信任。
- IdM 中已存在 AD 用户的用户 ID 覆盖。如果没有，使用 ipa idoverrideuser-add 'default trust view' *ad_user@ad.example.com* 命令创建它。
- 您要将用户 ID 覆盖添加至其中的组在 IdM 中已存在。
- 您可以使用 IdM 的 4.8.7 版本或更高版本。要查看您在服务器上安装的 IdM 版本，请输入 ipa --version。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 使用以下内容创建 add-useridoverride-to-group.yml playbook：

```
---
- name: Playbook to ensure presence of users in a group
  hosts: ipaserver


  - name: Ensure the ad_user@ad.example.com user ID override is a member of the
admins group:
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: admins
      idoverrideuser:
      - ad_user@ad.example.com
```

在示例中：

- Secret123 是 IdM 管理员密码。

- 管理员 是您要添加 ad_user@ad.example.com ID 覆盖的 IdM POSIX 组的名称。此组成员具有全部的管理员特权。

- ad_user@ad.example.com 是 AD 管理员的用户 ID 覆盖。用户存储在已建立信任的 AD 域中。

3. 保存该文件。

4. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-useridoverride-to-group.yml
```

其它资源

- AD 用户的 ID 覆盖
- /usr/share/doc/ansible-freeipa/README-group.md
- /usr/share/doc/ansible-freeipa/playbooks/user
- 在 Active Directory 环境中使用 ID 视图

## 87.4. 验证 AD 用户是否可以在 IDM CLI 中执行正确的命令

此流程检查 Active Directory(AD)用户可以登录到 Identity Management(IdM)命令行界面(CLI)，并运行适合其角色的命令。

1. 销毁 IdM 管理员的当前 Kerberos ticket：

```
# kdestroy -A
```

**注意**

Kerberos ticket 的破坏是必需的，因为 MIT Kerberos 中的 GSSAPI 实施首选项从目标服务域选择凭证，本例中为 IdM 域。这意味着，如果凭证缓存集合，即 KCM:、KEYRING:, or DIR: 凭证缓存类型在被使用，则之前获取的 admin 或其他 IdM 主体的凭证将用于访问 IdM API，而不是 AD 用户的凭证。

2. 获取已为其创建 ID 覆盖的 AD 用户的 Kerberos 凭证：

```
# kinit ad_user@AD.EXAMPLE.COM
Password for ad_user@AD.EXAMPLE.COM:
```

3. 测试 AD 用户的 ID 覆盖是否因 IdM 组成员资格而获得与该组中的任何 IdM 用户相同的权限。如果 AD 用户的 ID 覆盖已添加到 admins 组中，AD 用户可以在 IdM 中创建组：

```
# ipa group-add some-new-group
---------------------------
Added group "some-new-group"
---------------------------
  Group name: some-new-group
  GID: 1997000011
```

## 87.5. 使用 ANSIBLE 启用 AD 用户管理 IDM

您可以使用 ansible-freeipa idoverrideuser 和 group 模块从可信 AD 域中为活动目录(AD)用户创建用户 ID 覆盖，并为该用户授予与 IdM 用户相同的权限。该流程使用 Default Trust View ID 视图的示例，在第一个 playbook 任务中添加 administrator@addomain.com ID 覆盖。在下一个 playbook 任务中，administrator@addomain.com ID 覆盖作为成员添加到 IdM admins 组中。因此，AD 管理员可以管理 IdM，而无需两个不同的帐户和密码。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 您使用 RHEL 8.10 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- AD 林与 IdM 信任。在示例中，AD 域的名称是 addomain.com，AD 管理员的完全限定域名(FQDN)是 administrator@addomain.com。

- 清单文件中的 ipaserver 主机被配置为信任控制器或信任代理。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 在 Ansible 控制节点上，创建一个带有任务的 enable-ad-admin-to-administer-idm.yml playbook，将 administrator@addomain.com 用户覆盖添加到 Default Trust View 中：

```
---
- name: Enable AD administrator to act as a FreeIPA admin
  hosts: ipaserver
  become: false
  gather_facts: false

  tasks:
  - name: Ensure idoverride for administrator@addomain.com in 'default trust view'
    ipaidoverrideuser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      idview: "Default Trust View"
      anchor: administrator@addomain.com
```

2. 在同一 playbook 中使用另一个 playbook 任务，将 AD 管理员用户 ID 覆盖添加到 admins 组中：

```
  - name: Add the AD administrator as a member of admins
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: admins
      idoverrideuser:
      - administrator@addomain.com
```

3. 保存该文件。

4. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory enable-ad-admin-to-administer-idm.yml
```

验证

1. 以 AD Administrator 用户身份登录 IdM 客户端：

```
$ ssh administrator@addomain.com@client.idm.example.com
```

2. 验证您是否获得了有效的票据授予票(TGT)：

```
$ klist
Ticket cache: KCM:325600500:99540
Default principal: Administrator@ADDOMAIN.COM
Valid starting Expires Service principal
02/04/2024 11:54:16 02/04/2024 21:54:16 krbtgt/ADDOMAIN.COM@ADDOMAIN.COM
renew until 02/05/2024 11:54:16
```

3. 在 IdM 中验证您的 admin 权限：

```
$ ipa user-add testuser --first=test --last=user
----------------------
Added user "tuser"
----------------------
  User login: tuser
  First name: test
  Last name: user
  Full name: test user
[...]
```

其它资源

- idoverrideuser 和 ipagroup ansible-freeipa 上游文档
- 启用 AD 用户管理 IdM

# 第 88 章 配置域名解析顺序来解析较短的 AD 用户名

默认情况下，您必须指定格式为 user_name@domain.com 或 domain.com\user_name 的 完全限定名称，以便从 Active Directory(AD)环境中解析和验证用户和组。以下小节描述了如何配置 IdM 服务器和客户端来解析简短的 AD 用户名和组名称。

- 域解析顺序的工作方式
- 在 IdM 服务器中设置全局域解析顺序
- 为 IdM 服务器中的 ID 视图设置域解析顺序
- 使用 Ansible 创建 ID 视图，其域解析顺序
- 在 IdM 客户端上在 SSSD 中设置域解析顺序

## 88.1. 域解析顺序的工作方式

在具有 Active Directory(AD)信任的 Identity Management(IdM)环境中，红帽建议您通过指定完全限定名称来解析和验证用户和组。例如：

- *<idm_username>*@idm.example.com 适用于 idm.example.com 域中的 IdM 用户
- 用于 ad.example.com 域的 AD 用户的 *<ad_username>*@ad.example.com

默认情况下，如果您使用 *简短名称* 格式执行用户和组查找，如 ad_username，IdM 只会搜索 IdM 域，且无法找到 AD 用户和组。要使用短名称解析 AD 用户或组，请通过设置域 解析顺序选项来更改 IdM 搜索多个域的顺序。

您可以在 IdM 数据库或单个客户端的 SSSD 配置中设置域解析顺序。IdM 按照以下优先级顺序评估域解析顺序：

- 本地 /etc/sssd/sssd.conf 配置.
- ID 视图配置。
- 全局 IdM 配置。

备注

- 如果主机上的 SSSD 配置包含 default_domain_suffix 选项，并且您想要向未使用这个选项指定的域发出请求，则必须使用完全限定用户名。
- 如果您使用 域解析顺序 选项并查询 compat 树，您可能会收到多个用户 ID(UID)。如果这可能会影响您，请参阅 设置域解析顺序时 AD 用户的 Pagure 错误报告 Inconsistent compat 用户对象。

重要

不要在 IdM 客户端或 IdM 服务器中使用 full_name_format SSSD 选项。为这个选项使用非默认值会更改用户名的显示方式，并可能会破坏 IdM 环境中的查找。

其它资源

- 传统 Linux 客户端的活动目录信任.

## 88.2. 在 IDM 服务器中设置全局域解析顺序

此流程为 IdM 域中的所有客户端设置域解析顺序。这个示例按以下顺序设置搜索用户和组的域解析顺序：

1. Active Directory(AD)root 域 ad.example.com
2. AD 子域 子域1.ad.example.com

3. IdM 域 idm.example.com

**先决条件**

- 您已使用 AD 环境配置了信任关系。

**流程**

- 使用 ipa config-mod --domain-resolution-order 命令列出按您首选顺序搜索的域。使用冒号(:)分隔域。

```
[user@server ~]$ ipa config-mod --domain-resolution-
order='ad.example.com:subdomain1.ad.example.com:idm.example.com'
Maximum username length: 32
Home directory base: /home
...
  Domain Resolution Order:
ad.example.com:subdomain1.ad.example.com:idm.example.com
...
```

验证步骤

- 验证您是否可以仅使用简短名称从 ad.example.com 域检索用户的用户信息。

```
[root@client ~]# id <ad_username>
uid=1916901102(ad_username) gid=1916900513(domain users)
groups=1916900513(domain users)
```

### 88.3. 为 IDM 服务器中的 ID 视图设置域解析顺序

此流程为可应用于一组特定 IdM 服务器和客户端的 ID 视图设置域解析顺序。这个示例为 IdM 主机 client1.idm.example.com 创建名为 ADsubdomain1_first 的 ID 视图，并设置按照以下顺序搜索用户和组的域解析顺序：

1. Active Directory(AD)子域 subdomain1.ad.example.com

2. AD root 域 ad.example.com

3.

IdM 域 idm.example.com

注意

ID 视图中设置的域解析顺序覆盖全局域解析顺序，但它不会覆盖 SSSD 配置在本地设置的任何域解析顺序。

先决条件

- 您已使用 AD 环境配置了信任关系。

流程

1. 创建 ID 视图，并设置 --domain-resolution-order 选项。

```
[user@server ~]$ ipa idview-add ADsubdomain1_first --desc "ID view for resolving AD
subdomain1 first on client1.idm.example.com" --domain-resolution-order
subdomain1.ad.example.com:ad.example.com:idm.example.com
---------------------------------
Added ID View "ADsubdomain1_first"
---------------------------------
ID View Name: ADsubdomain1_first
Description: ID view for resolving AD subdomain1 first on client1.idm.example.com
Domain Resolution Order:
subdomain1.ad.example.com:ad.example.com:idm.example.com
```

2. 将 ID 视图应用到 IdM 主机。

```
[user@server ~]$ ipa idview-apply ADsubdomain1_first --hosts
client1.idm.example.com
----------------------------------
Applied ID View "ADsubdomain1_first"
----------------------------------
  hosts: client1.idm.example.com
------------------------------------------
Number of hosts the ID View was applied to: 1
------------------------------------------
```

验证步骤

- 显示 ID 视图的详细信息。

```
[user@server ~]$ ipa idview-show ADsubdomain1_first --show-hosts
  ID View Name: ADsubdomain1_first
  Description: ID view for resolving AD subdomain1 first on client1.idm.example.com
  Hosts the view applies to: client1.idm.example.com
  Domain resolution order:
subdomain1.ad.example.com:ad.example.com:idm.example.com
```

- 验证您只能使用简短名称从 subdomain1.ad.example.com 域检索用户的用户信息。

```
[root@client1 ~]# id <user_from_subdomain1>
uid=1916901106(user_from_subdomain1) gid=1916900513(domain users)
groups=1916900513(domain users)
```

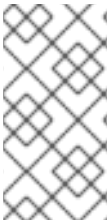
## 88.4. 使用 ANSIBLE 创建 ID 视图，其域解析顺序

您可以使用 ansible-freeipa idview 模块在 Identity Management (IdM)部署中添加、修改和删除 ID 视图。例如，您可以使用域解析顺序创建 ID 视图来启用简短名称表示法。

短名称表示法从 Active Directory (AD)替换完整的用户名，如 aduser05@ad.example.com，并带有短登录信息，本例中为 aduser05。这意味着，当使用 SSH 登录到 IdM 客户端时，aduser05 可以输入 ssh aduser05@client.idm.example.com 而不是 ssh aduser05@ad.example.com@client.idm.example.com。这同样适用于其他命令，如 id。

完成此流程以使用 Ansible：

- 定义用于短名称资格的冒号分隔域字符串。在示例中，字符串是 ad.example.com:idm.example.com。
- 创建一个 ID 视图，以指示 SSSD 首先在字符串中标识的第一个域中搜索用户名。在示例中，这是 ad.example.com。
- 将 ID 视图应用到特定的主机。在示例中，这是 testhost.idm.example.com。



**注意**

您只能将一个 ID 视图应用到 IdM 客户端。应用新的 ID 视图（如果适用）会自动删除以前的 ID 视图。

**先决条件**

- 在控制节点上：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已安装 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您使用 RHEL 8.10 及更新的版本。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。
- testhost.idm.example.com 是一个 IdM 客户端。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入 ~/**MyPlaybooks**/ 目录，并使用以下内容创建一个 Ansible playbook 文件 add-id-view-with-domain-resolution-order.yml ：

```
---
- name: Playbook to add idview and apply it to an IdM client
  hosts: ipaserver
  vars_files:
  - /home/<user_name>/MyPlaybooks/secret.yml
  become: false
  gather_facts: false

  tasks:
  - name: Add idview and apply it to testhost.idm.example.com
    ipaidview:
      ipaadmin_password: "{{ ipaadmin_password }}"
```

```
    name: test_idview
    host: testhost.idm.example.com
    domain_resolution_order: "ad.example.com:ipa.example.com"
```

2. 运行 playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-id-view-with-domain-resolution-order.yml
```

验证

1. SSH 到 testhost.idm.example.com。

2. 验证您只能使用短名称从 ad.example.com 域检索用户的用户信息。

```
[root@testhost ~]# id aduser05
uid=1916901102(aduser05) gid=1916900513(domain users) groups=1916900513(domain users)
```

其它资源

- ansible-freeipa 上游文档中的 idview 模块

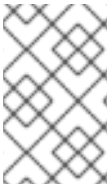
## 88.5. 在 IDM 客户端上在 SSSD 中设置域解析顺序

此流程在 IdM 客户端上的 SSSD 配置中设置域解析顺序。这个示例将 IdM 主机 client2.idm.example.com 配置为按以下顺序搜索用户和组：

1. Active Directory(AD)子域 subdomain1.ad.example.com

2. AD root 域 ad.example.com

3. IdM 域 idm.example.com



**注意**

本地 SSSD 配置中的域解析顺序覆盖任何全局和 ID 视图域解析顺序。

**先决条件**

- 您已使用 AD 环境配置了信任关系。

**流程**

1. 在文本编辑器中打开 /etc/sssd/sssd.conf 文件。
2. 在文件的 [sssd] 部分中设置 domain_resolution_order 选项。

```
domain_resolution_order = subdomain1.ad.example.com, ad.example.com,
idm.example.com
```

3. 保存并关闭该文件。
4. 重启 SSSD 服务以加载新的配置设置。

```
[root@client2 ~]# systemctl restart sssd
```

**验证步骤**

- 验证您只能使用简短名称从 subdomain1.ad.example.com 域检索用户的用户信息。

```
[root@client2 ~]# id <user_from_subdomain1>
uid=1916901106(user_from_subdomain1) gid=1916900513(domain users)
groups=1916900513(domain users)
```

## 88.6. 其它资源

- 使用 ID 视图来覆盖 IdM 客户端上的用户属性值

# 第 89 章 在 IDM 中使用 AD 用户主体名称启用身份验证

## 89.1. IDM 信任的 AD 林中的用户主体名称

作为身份管理(IdM)管理员，您可以允许 AD 用户使用其他用户主体名称 (UPN)访问 IdM 域中的资源。UPN 是 AD 用户以 user_name@KERBEROS-REALM 格式通过 进行身份验证的替代用户登录。作为 AD 管理员，您可以为 user_name 和 KERBEROS-REALM 设置备选值，因为您可以在 AD 林中配置额外的 Kerberos 别名和 UPN 后缀。

例如，如果公司使用 Kerberos 域 AD.EXAMPLE.COM，用户的默认 UPN 为 user@ad.example.com。要允许您的用户使用其电子邮件地址（如 user@example.com ）登录，您可以在 AD 中将 EXAMPLE.COM 配置为替代的 UPN。如果贵公司最近进行了合并，并且希望为用户提供统一的登录命名空间，备选 *UPN（也称为企业 UPN）*特别方便。

只有在 AD 林根目录中定义时，UPN 后缀才对 IdM 可见。作为 AD 管理员，您可以使用 Active Directory 域和 Trust utility 或 PowerShell 命令行工具来定义 UPN。

**注意**

要为用户配置 UPN 后缀，红帽建议使用执行错误验证的工具，如 Active Directory 域和 Trust 实用程序。

红帽建议不要通过低级修改来配置 UPN，例如使用 ldapmodify 命令为用户设置 userPrincipalName 属性，因为 Active Directory 不验证这些操作。

在 AD 端定义一个新的 UPN 后，在 IdM 服务器中运行 ipa trust-fetch-domains 命令以检索更新的 UPN。请参阅确保 AD UPN 在 IdM 中是最新的。

IdM 将域的 UPN 后缀存储在子树 cn=trusted_domain_name,cn=ad,cn=trusts,dc=idm,dc=example,dc=com 的多值属性 ipaNTAdditionalSuffixes 中。

**其它资源**

- 如何在 AD 林根目录中编写 UPN 后缀设置脚本

- 如何手动修改 AD 用户条目并绕过任何 UPN 后缀验证
- 信任控制器和信任代理

## 89.2. 确保 AD UPN 在 IDM 中是最新的

在可信 Active Directory(AD)林中添加或删除用户主体名称(UPN)后缀后，刷新 IdM 服务器上的可信林的信息。

**先决条件**

- IdM 管理员凭证。

**流程**

- 输入 ipa trust-fetch-domains 命令。请注意，预计会出现一个看似为空的输出：

```
[root@ipaserver ~]# ipa trust-fetch-domains
Realm-Name: ad.example.com
-----------------------------
No new trust domains were found
-----------------------------
----------------------------
Number of entries returned 0
----------------------------
```

验证步骤

- 输入 ipa trust-show 命令，以验证服务器是否已获取新的 UPN。在提示时指定 AD 域的名称：

```
[root@ipaserver ~]# ipa trust-show
Realm-Name: ad.example.com
  Realm-Name: ad.example.com
  Domain NetBIOS name: AD
  Domain Security Identifier: S-1-5-21-796215754-1239681026-23416912
  Trust direction: One-way trust
  Trust type: Active Directory domain
  UPN suffixes: example.com
```

输出显示 example.com UPN 后缀现在是 ad.example.com 域条目的一部分。

## 89.3. 为 AD UPN 身份验证问题收集故障排除数据

按照以下流程，从活动目录(AD)环境和 IdM 环境收集有关用户主体名称(UPN)配置的故障排除数据。如果您的 AD 用户无法使用备用 UPN 登录，您可以使用此信息缩小故障排除工作范围。

**先决条件**

- 您必须登录到 IdM Trust Controller 或 Trust Agent，才能从 AD 域控制器检索信息。
- 您需要 root 权限才能修改以下配置文件，并重新启动 IdM 服务。

**流程**

1. 在文本编辑器中打开 /usr/share/ipa/smb.conf.empty 配置文件。
2. 将以下内容添加到该文件中。

```
[global]
log level = 10
```

3. 保存并关闭 /usr/share/ipa/smb.conf.empty 文件。
4. 在文本编辑器中打开 /etc/ipa/server.conf 配置文件。如果没有该文件，请创建一个。
5. 将以下内容添加到该文件中。

```
[global]
debug = True
```

6. 保存并关闭 /etc/ipa/server.conf 文件。

7. 重启 Apache webserver 服务以应用配置更改：

```
[root@server ~]# systemctl restart httpd
```

8. 从您的 AD 域检索信任信息：

```
[root@server ~]# ipa trust-fetch-domains <ad.example.com>
```

9. 查看以下日志文件中的调试输出和故障排除信息：

- /var/log/httpd/error_log
- /var/log/samba/log.*

**其它资源**

- 请参阅 使用 rpcclient 来收集 AD UPN 身份验证方面问题的故障排除数据。

# 第 90 章 在 IDM 中使用规范化 DNS 主机名

默认情况下，在 Identity Management(IdM)客户端上禁用 DNS 规范化，以避免潜在的安全风险。例如，如果攻击者控制 DNS 服务器和域中的主机，攻击者可以导致短主机名（如 demo ）解析到被入侵的主机，如 bad .example.com。在这种情况下，用户连接到与预期不同的服务器。

这个流程描述了如何在 IdM 客户端中使用规范化主机名。

## 90.1. 向主机主体中添加别名

默认情况下，使用 ipa-client-install 命令注册的身份管理(IdM)客户端不允许在服务主体中使用短主机名。例如，在访问服务时，用户只能使用 host/demo.example.com@EXAMPLE.COM ，而不是 host/demo@EXAMPLE.COM。

按照以下流程在 Kerberos 主体中添加别名。请注意，您也可以在 /etc/krb5.conf 文件中启用主机名规范化。详情请参阅 在客户端上的服务主体中启用主机名规范。

**先决条件**

- 已安装 IdM 客户端。
- 主机名在网络中是唯一的。

**流程**

1. 以 admin 用户身份对 IdM 进行身份验证：

```
$ kinit admin
```

2. 将别名添加到主机主体。例如，要在 demo. examle.com 主机主体中添加 demo 别名：

```
$ ipa host-add-principal demo.example.com --principal=demo
```

## 90.2. 在客户端的服务主体中启用主机名规范

按照以下流程，在客户端上的服务主体中启用主机名规范化。

请注意，如果您使用主机主体别名，如 将别名添加到主机主体 中所述，则不需要启用规范。

**先决条件**

- 已安装 Identity Management(IdM)客户端。
- 以 root 用户身份登录 IdM 客户端。
- 主机名在网络中是唯一的。

**流程**

1. 将 /etc/krb5.conf 文件中的 [libdefaults] 部分中的 dns_canonicalize_hostname 参数设置为 false ：

```
[libdefaults]
...
dns_canonicalize_hostname = true
```

## 90.3. 启用 DNS 主机名规范化使用主机名的选项

如果您在 /etc/krb5.conf 文件中设置了 dns_canonicalize_hostname = true，如 在客户端上的服务主体中启用主机名规范 中所述，在服务主体中使用主机名时，您有如下选择：

- 在 Identity Management(IdM)环境中，您可以在服务主体中使用完整主机名，如 host/demo.example.com@EXAMPLE.COM。
- 在没有 IdM 的环境中，但如果 RHEL 主机作为 Active Directory(AD)域的成员，则不需要进一步考虑，因为 AD 域控制器(DC)自动为注册到 AD 的机器的 NetBIOS 名称创建服务主体。

# 第 91 章 使用 ANSIBLE PLAYBOOK 在 IDM 中管理全局 DNS 配置

使用 Red Hat Ansible Engine dnsconfig 模块，您可以配置 Identity Management(IdM)DNS 的全局配置。全局 DNS 配置中定义的设置应用到所有 IdM DNS 服务器。但是，全局配置优先于特定 IdM DNS 区的配置。

dnsconfig 模块支持以下变量：

- 全局转发器，特别是 IP 地址和用于通信的端口。
- 全局转发策略：只有、first 或 none。有关这些 DNS 转发策略类型的详情，请查看 IdM 中的 DNS 转发策略。
- 同步正向查找和反向查找区域。

先决条件

- DNS 服务安装在 IdM 服务器上。有关如何使用集成 DNS 安装 IdM 服务器的详情，请查看以下链接之一：
  - 安装 IdM 服务器：使用集成的 DNS，集成的 CA 作为 root CA
  - 安装 IdM 服务器：具有集成的 DNS，具有外部 CA 作为根CA
  - 安装 IdM 服务器：使用集成的 DNS,没有 CA

本章包括以下部分：

- IdM 如何确保 NetworkManager 不会删除 /etc/resolv.conf 中的全局转发器

- 使用 Ansible 在 IdM 中存在 DNS 全局转发器
- 使用 Ansible 确保 IdM 中没有 DNS 全局转发器
- ipadnsconfig ansible-freeipa 模块中的 action: member 选项
- IdM 中 DNS 转发策略的介绍
- 使用 Ansible playbook 确保 IdM DNS 全局配置中设置了 forward first 策略
- 使用 Ansible playbook 确保 IdM DNS 中禁用了全局转发器
- 使用 Ansible playbook 确保 IdM DNS 中禁用了正向和反向查找区域的同步

## 91.1. IDM 如何确保 NETWORKMANAGER 不会删除 /ETC/RESOLV.CONF 中的全局转发器

使用集成 DNS 安装 Identity Management(IdM)将 /etc/resolv.conf 文件配置为指向 127.0.0.1 localhost 地址：

```
# Generated by NetworkManager
search idm.example.com
nameserver 127.0.0.1
```

在某些情况下，如使用 动态主机配置 协议(DHCP)的网络，NetworkManager 服务可能会恢复对 /etc/resolv.conf 文件的更改。为了使 DNS 配置持久，IdM DNS 安装过程还通过以下方式配置 NetworkManager 服务：

1. DNS 安装脚本会创建一个 /etc/NetworkManager/conf.d/zzz-ipa.conf NetworkManager 配置文件来控制搜索顺序和 DNS 服务器列表：

```
# auto-generated by IPA installer
[main]
dns=default

[global-dns]
```

```
searches=$DOMAIN

[global-dns-domain-*]
servers=127.0.0.1
```

2. NetworkManager 服务已重新加载，该服务始终使用 /etc/NetworkManager/conf.d/ 目录中的最后一个文件中的设置来创建 /etc/resolv.conf 文件。这时为 zzz-ipa.conf 文件。

重要

不要手动修改 /etc/resolv.conf 文件。

## 91.2. 使用 ANSIBLE 在 IDM 中存在 DNS 全局转发器

按照以下流程，使用 Ansible playbook 确保 DNS 全局转发器在 IdM 中存在。在下例中，IdM 管理员确保在 DNS 服务器中存在一个到 IPv4 地址为 7.7.9.9，IP v6 地址为 2001:db8::1:0，端口 53 的 DNS global forwarder。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 制作 forwarders-absent.yml Ansible playbook 文件的副本。例如：

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. 打开 ensure-presence-global-forwarder.yml 文件进行编辑。

5. 通过设置以下变量来调整文件：

   a. 将 playbook 的 name 变量 更改为 Playbook，以确保 IdM DNS 中存在全局转发器。

   b. 在 tasks 部分中，将任务 的名称 更改为确保 存在 DNS global forwarder 在端口 53 上存在 7.7.9.9 和 2001:db8::1:0。

   c. 在 ipadnsconfig 部分的 forwarders 部分：

      i. 将第一个 ip_address 值更改为全局转发器的 IPv4 地址： 7.7.9.9。

      ii. 将第二个 ip_address 值更改为全局转发器的 IPv6 地址： 2001:db8::1:0。

      iii. 验证 端口 值是否已设置为 53。

.

d.

将 状态 更改为 present。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Playbook to ensure the presence of a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0
on port 53
    ipadnsconfig:
      forwarders:
        - ip_address: 7.7.9.9
        - ip_address: 2001:db8::1:0
          port: 53
      state: present
```

6.

保存该文件。

7.

运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-
presence-of-a-global-forwarder.yml
```

其它资源

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsconfig.md 文件。

### 91.3. 使用 ANSIBLE 确保 IDM 中没有 DNS 全局转发器

按照以下流程，使用 Ansible playbook 确保 DNS 全局转发器在 IdM 中不存在。在以下示例流程中，IdM 管理员确保在端口 53 上没有互联网协议(IP)v4 地址为 8.8.6.6 和 IP v6 地址为 2001:4860:4860::8800 的 DNS 全局转发器。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：

- 您使用 Ansible 版本 2.14 或更高版本。
- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 制作 forwarders-absent.yml Ansible playbook 文件的副本。例如：

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. 打开 ensure-absence-of-a-global-forwarder.yml 文件进行编辑。

5. 通过设置以下变量来调整文件：

a. 将 playbook 的 name 变量 更改为 Playbook，以确保 IdM DNS 中缺少全局转发器。

b. 在 tasks 部分中，将任务 的名称 更改为确保 没有 DNS 全局转发器在端口 53 上为 8.8.6.6 和 2001:4860:4860::8800。

c. 在 ipadnsconfig 部分的 forwarders 部分：

i. 将第一个 ip_address 值更改为全局转发器的 IPv4 地址： 8.8.6.6。

ii. 将第二个 ip_address 值更改为全局转发器的 IPv6 地址： 2001:4860:4860::8800。

iii. 验证 端口 值是否已设置为 53。

d. 将 action 变量设置为 member。

e. 验证 state 已设为 absent。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Playbook to ensure the absence of a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the absence of a DNS global forwarder to 8.8.6.6 and
2001:4860:4860::8800 on port 53
    ipadnsconfig:
      forwarders:
        - ip_address: 8.8.6.6
        - ip_address: 2001:4860:4860::8800
          port: 53
      action: member
      state: absent
```

**重要**

如果您仅在 playbook 中使用 state: absent 选项，而不使用 action: member，则 playbook 会失败。

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

**其它资源**

- /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsconfig.md 文件
- ipadnsconfig ansible-freeipa 模块中的 action: member 选项

## 91.4. IPADNSCONFIG ANSIBLE-FREEIPA 模块中的 ACTION: MEMBER 选项

使用 ansible-freeipa ipadnsconfig 模块在身份管理(IdM)中排除全局转发器，除了使用 state: absent 选项外，还需要使用 action: member 选项。如果您只使用 playbook 中的 state: absent，而没有使用 action: member，则 playbook 将失败。因此，要删除所有全局转发器，您必须在 playbook 中单独指定它们。相反，state: present 选项不需要 action: member。

下表 提供了添加和删除 DNS 全局转发器的配置示例，其演示了 action: member 选项的正确使用。表中每一行显示了：

- 执行 playbook 前配置的全局转发器
- playbook 摘录
- 执行 playbook 后配置的全局转发器

**表 91.1. 全局转发器的 ipadnsconfig 管理**

| 之前的转发器 | Playbook 摘录 | 之后的转发器 |
| --- | --- | --- |
| 8.8.6.6 | [...]<br>tasks:<br>- name: Ensure the presence of DNS global forwarder 8.8.6.7<br>ipadnsconfig:<br>forwarders:<br>- ip_address: 8.8.6.7<br>state: present | 8.8.6.7 |
| 8.8.6.6 | [...]<br>tasks:<br>- name: Ensure the presence of DNS global forwarder 8.8.6.7<br>ipadnsconfig:<br>forwarders:<br>- ip_address: 8.8.6.7<br>action: member<br>state: present | 8.8.6.6, 8.8.6.7 |
| 8.8.6.6, 8.8.6.7 | [...]<br>tasks:<br>- name: Ensure the absence of DNS global forwarder 8.8.6.7<br>ipadnsconfig:<br>forwarders:<br>- ip_address: 8.8.6.7<br>state: absent | 尝试执行 playbook 会导致错误。原始配置 - 8.8.6.6、8.8.6.7 - 保持不变。 |
| 8.8.6.6, 8.8.6.7 | [...]<br>tasks:<br>- name: Ensure the absence of DNS global forwarder 8.8.6.7<br>ipadnsconfig:<br>forwarders:<br>- ip_address: 8.8.6.7<br>action: member<br>state: absent | 8.8.6.6 |

### 91.5. IDM 中的 DNS 转发策略

**IdM 支持 第一个 且 唯一的 标准 BIND 转发策略，以及任何 IdM 特定的转发策略。**

**首先转发 *（默认）***

**IdM BIND 服务将 DNS 查询转发到配置的转发器。如果因为服务器错误或超时查询失败，BIND**

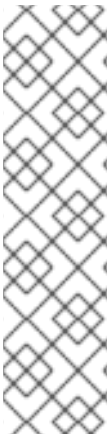
会使用 Internet 上的服务器回退到递归解析。forward first 策略是默认策略，它适合优化 DNS 流量。

仅转发

IdM BIND 服务将 DNS 查询转发到配置的转发器。如果因为服务器错误或超时查询失败，BIND 会向客户端返回错误。建议在带有分割 DNS 配置的环境中使用 forward only 策略。

**none** *（转发禁用）*

DNS 查询不会通过 none 转发策略转发。禁用转发仅作为全局转发配置的特定区覆盖。这个选项等同于在 BIND 配置中指定空转发器列表。



**注意**

您不能使用转发将 IdM 中的数据与其他 DNS 服务器的数据组合。您只能在 IdM DNS 中转发主区的查询。

默认情况下，如果查询的 DNS 名称属于 IdM 服务器具有权威的区域，BIND 服务不会将查询转发到另一服务器。在这种情况下，如果无法在 IdM 数据库中找到查询的 DNS 名称，则会返回 NXDOMAIN 回答。未使用转发。

**例 91.1. 场景示例**

IdM 服务器对 test.example 具有权威。DNS 区域.BIND 配置为将查询转发到 IP 地址 192.0.2.254 的 DNS 服务器。

客户端发送对 不存在.test.example的查询时。DNS 名称，BIND 检测到 IdM 服务器对 test.example. 区域具有权威，并且不会将查询转发到 192.0.2.254. 服务器。因此，DNS 客户端会收到 NXDomain 错误消息，通知用户查询的域不存在。

## 91.6. 使用 ANSIBLE PLAYBOOK 确保 IDM DNS 全局配置中设置了 FORWARD FIRST 策略

按照以下流程，使用 Ansible playbook 确保 IdM DNS 中的全局转发策略被设置为 forward first。

如果您使用 forward first DNS 转发策略，DNS 查询将转发到配置的转发器。如果因为服务器错误或超时查询失败，BIND 会使用 Internet 上的服务器回退到递归解析。forward first 策略是默认策略。它适用于流量优化。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。
- 您的 IdM 环境包含一个集成的 DNS 服务器。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 复制 set-configuration.yml Ansible playbook 文件。例如：

```
$ cp set-configuration.yml set-forward-policy-to-first.yml
```

4. 打开 set-forward-policy-to-first.yml 文件进行编辑。

5. 通过在 ipadnsconfig task 部分中设置以下变量来调整文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员密码。
   - 将 forward_policy 变量设置为 first。

   删除原始 playbook 的其他所有行。这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to set global forwarding policy to first
  hosts: ipaserver
  become: true

  tasks:
  - name: Set global forwarding policy to first.
    ipadnsconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      forward_policy: first
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file set-forward-policy-to-first.yml
```

**其它资源**

- 请参阅 IdM 中的 DNS 转发策略。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsconfig.md 文件。
- 如需更多 playbook 示例，请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目

录。

## 91.7. 使用 ANSIBLE PLAYBOOK 确保 IDM DNS 中禁用了全局转发器

按照以下流程，使用 Ansible playbook 确保全局转发器在 IdM DNS 中被禁用了。禁用的方法是将 forward_policy 变量设置为 none。

禁用全局转发器会导致 DNS 查询不会被转发。禁用转发仅作为全局转发配置的特定区覆盖。这个选项等同于在 BIND 配置中指定空转发器列表。

### 先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。
- 您的 IdM 环境包含一个集成的 DNS 服务器。

### 流程

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 disable-global-forwarders.yml Ansible playbook 文件的副本。例如：

```
$ cp disable-global-forwarders.yml disable-global-forwarders-copy.yml
```

4. 打开 disable-global-forwarders-copy.yml 文件进行编辑。

5. 通过在 ipadnsconfig task 部分中设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 forward_policy 变量设置为 none。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to disable global DNS forwarders
  hosts: ipaserver
  become: true

  tasks:
  - name: Disable global forwarders.
    ipadnsconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      forward_policy: none
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disable-global-forwarders-copy.yml
```

其它资源

- 请参阅 IdM 中的 DNS 转发策略。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsconfig.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录中的更多 playbook 示例。

## 91.8. 使用 ANSIBLE PLAYBOOK 确保 IDM DNS 中禁用了正向和反向查找区域的同步

按照以下流程，使用 Ansible playbook 确保正向和反向查找区域在 IdM DNS 中未同步。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

- 您的 IdM 环境包含一个集成的 DNS 服务器。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 复制 disallow -reverse-sync.yml Ansible playbook 文件。例如：

```
$ cp disallow-reverse-sync.yml disallow-reverse-sync-copy.yml
```

4. 打开 disallow -reverse-sync-copy.yml 文件进行编辑。

5. 通过在 ipadnsconfig task 部分中设置以下变量来调整文件：

   - 将 ipaadmin_password 变量设置为 IdM 管理员密码。
   - 将 allow_sync_ptr 变量设置为 no。

     这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to disallow reverse record synchronization
  hosts: ipaserver
  become: true

  tasks:
  - name: Disallow reverse record synchronization.
    ipadnsconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      allow_sync_ptr: no
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disallow-reverse-sync-copy.yml
```

**其它资源**

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsconfig.md 文件。
- 如需更多 playbook 示例，请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录。

# 第 92 章 管理 IDM 中的 DNS 区域

作为身份管理(IdM)管理员，您可以管理 IdM DNS 区域的工作方式。本章描述了以下主题和程序：

- IdM 中支持哪些 DNS 区域类型
  - 如何使用 IdM Web UI 添加主要 IdM DNS 区域
  - 如何使用 IdM CLI 添加主要 IdM DNS 区域
  - 如何使用 IdM Web UI 删除主 IdM DNS 区域
  - 如何使用 IdM CLI 删除主要 IdM DNS 区域
- 您可以在 IdM 中配置的 DNS 属性
  - 您如何在 IdM Web UI 中配置这些属性
  - 您如何在 IdM CLI 中配置这些属性
- IdM 中的区传送工作
  - 如何允许 IdM Web UI 中的区域传送
  - 如何允许 IdM CLI 中的区域传送

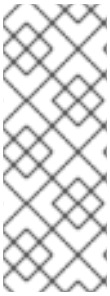
**先决条件**

- DNS 服务安装在 IdM 服务器上。有关如何使用集成 DNS 安装 IdM 服务器的详情，请查看以下链接之一：

- 安装 IdM 服务器：使用集成的 DNS，集成的 CA 作为 root CA
- 安装 IdM 服务器：具有集成的 DNS，具有外部 CA 作为根CA
- 安装 IdM 服务器：使用集成的 DNS,没有 CA

## 92.1. 支持的 DNS 区类型

身份管理 (IdM) 支持两种类型的 DNS 区域：*primary* 和 *forward*区域。此处描述了这两种类型的区，包括 DNS 转发的示例场景。



**注意**

本指南对区域类型使用 BIND 术语，不同于用于 Microsoft Windows DNS 的术语。BIND 服务器中的 Primary zones 与 Microsoft Windows DNS 中的 *forward lookup zones* 和 *reverse lookup zones* 作用相同。BIND 中的转发区域的作用与 Microsoft Windows DNS *中的条件* 转发器相同。

### 主 DNS 区域

主 DNS 区域包含权威 DNS 数据，并且可以接受动态 DNS 更新。此行为等同于标准 BIND 配置中的 类型 master 设置。您可以使用 ipa dnszone-* 命令管理主区域。

根据标准 DNS 规则，每个主区域必须包含 授权起始 (SOA)和 名称服务器 (NS)记录。在创建 DNS 区域时，IdM 会自动生成这些记录，但您必须手动将 NS 记录复制到父区域，以创建正确委托。

根据标准 BIND 行为，查询服务器不具有权威的名称将转发到其他 DNS 服务器。这些 DNS 服务器（称为转发器）可能是也可能不是查询的权威。

**例 92.1. DNS 转发的示例**

IdM 服务器包含 test.example. 主区域。此区域包含 sub.test.example. 名称的 NS 委派记录。此外，test.example. 区域为sub.text.example 子区域配置了 192.0.2.254 转发器 IP 地址。

查询名称 不存在.test.example. 的客户端会收到 NXDomain 回答，并且不会发生转发，因为 IdM 服务器对此名称具有权威。

另一方面，查询 host1.sub.test.example. name 会转发到配置的转发器 192.0.2.254，因为 IdM 服务器对此名称没有权威。

转发 DNS 区域

从 IdM 的角度来看，转发 DNS 区域不包含任何权威数据。事实上，正向"区"通常仅包含两段信息：

- 域名
- 与域关联的 DNS 服务器的 IP 地址

所有对属于定义的域的名称的查询都转发到指定的 IP 地址。此行为等同于标准 BIND 配置中的 type forward 设置。您可以使用 ipa dnsforwardzone-* 命令管理转发区。

转发 DNS 区域在 IdM-Active Directory(AD)信任的上下文中特别有用。如果 IdM DNS 服务器对 idm.example.com 区域具有权威，并且 AD DNS 服务器对 ad.example.com 区域具有权威，则 ad.example.com 是 idm.example.com 主区的 DNS 转发区域。这意味着，当查询来自 IdM 客户端以获取 somehost.ad.example.com 的 IP 地址时，查询将转发到 ad.example.com IdM DNS 转发区域中指定的 AD 域控制器。

## 92.2. 在 IDM WEB UI 中添加主 DNS 区域

按照以下流程，使用身份管理(IdM) Web UI 添加主 DNS 区域。

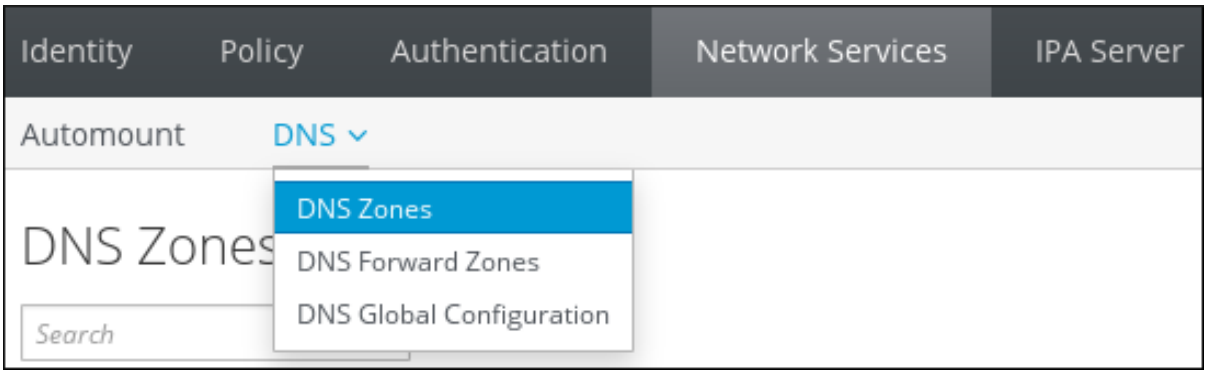
先决条件

- 以 IdM 管理员身份登录。

流程

1. 在 IdM Web UI 中，点击 Network Services → DNS → DNS Zones。

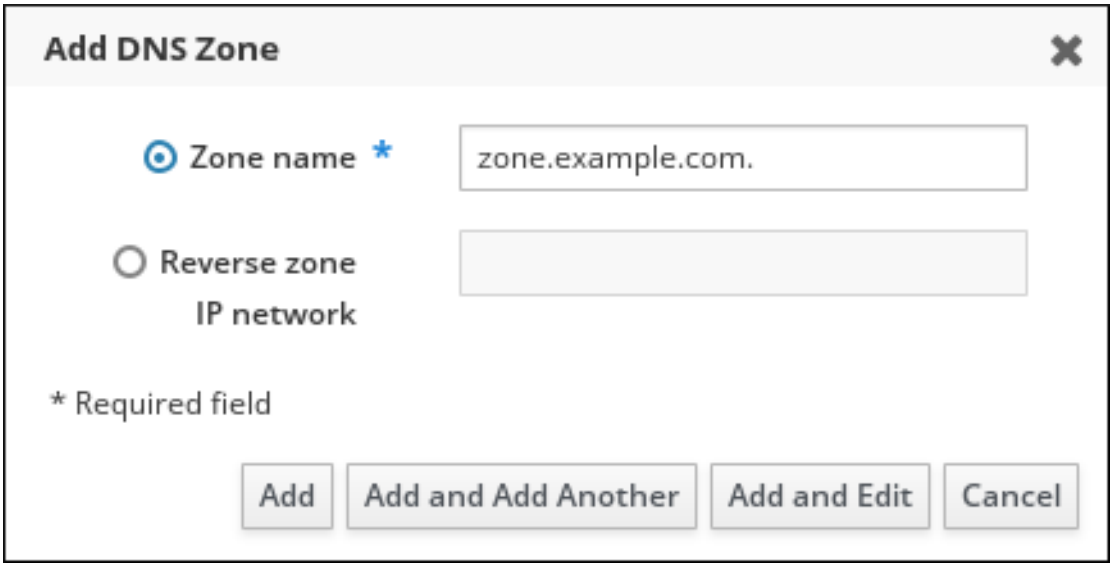
图 92.1. 管理 IdM DNS 主区域




2. 单击所有区域列表顶部的 Add。

3. 提供区域名称。

图 92.2. 输入一个新的 IdM 主区




4. 点添加。

## 92.3. 在 IDM CLI 中添加主 DNS 区域

按照以下流程，使用身份管理(IdM)命令行界面(CLI)添加主 DNS 区域。

### 先决条件

- 以 IdM 管理员身份登录。

**流程**

- ipa dnszone-add 命令向 DNS 域添加新区域。添加新区域要求您指定新子域的名称。您可以直接使用以下命令传递子域名：

```
$ ipa dnszone-add newzone.idm.example.com
```

如果没有将名称传递给 ipa dnszone-add，脚本会自动提示它。

**其它资源**

- 请参阅 ipa dnszone-add --help。

## 92.4. 在 IDM WEB UI 中删除主 DNS 区域

按照以下流程，使用 IdM Web UI 从身份管理(IdM)中删除主 DNS 区域。

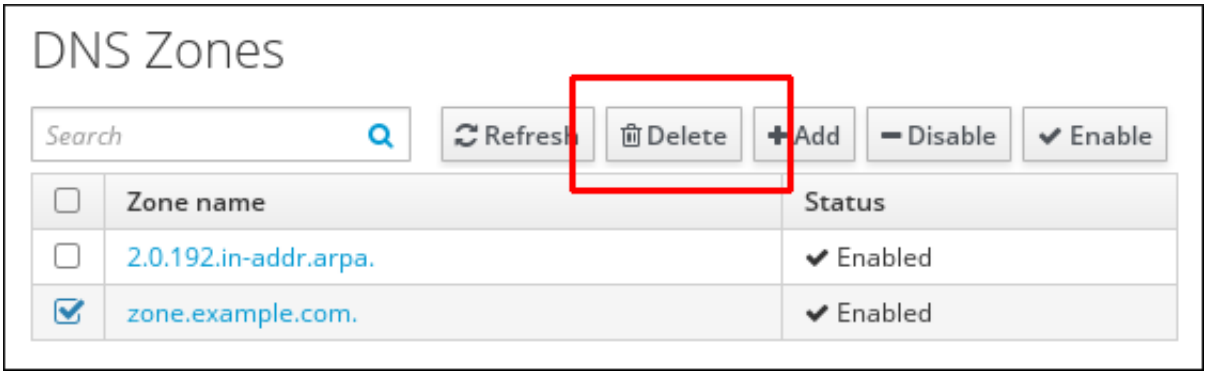
**先决条件**

- 以 IdM 管理员身份登录。

**流程**

1. 在 IdM Web UI 中，点击 Network Services → DNS → DNS Zones。
2. 按区域名称选择复选框，然后单击 Delete。

图 92.3. 删除主 DNS 区域



3. 在 Remove DNS 区域 对话框窗口中，确认您要删除所选区域。

## 92.5. 在 IDM CLI 中删除主 DNS 区域

按照以下流程，使用 IdM 命令行界面(CLI)从身份管理(IdM)中删除主 DNS 区域。

**先决条件**

- 以 IdM 管理员身份登录。

**流程**

- 要删除主 DNS 区域，输入 ipa dnszone-del 命令，后跟您要删除的区域的名称。例如：

```
$ ipa dnszone-del idm.example.com
```

## 92.6. DNS 配置优先级

您可以在以下层面上配置多个 DNS 配置选项：每个级别具有不同的优先级。

**特定于区域的配置**

IdM 中定义的特定区的具体配置级别具有最高的优先级。您可以使用 ipa dnszone-* 和 ipa dnsforwardzone-* 命令来管理特定于区的配置。

**每服务器配置**

安装 IdM 服务器过程中，需要您定义每服务器转发器。您可以使用 ipa dnsserver-* 命令管理每服务器转发器。如果您不想在安装副本时设置每服务器转发器，您可以使用 --no-forwarder 选项。

**全局 DNS 配置**

如果没有定义特定于区域的配置，IdM 将使用存储在 LDAP 中的全局 DNS 配置。您可以使用 ipa dnsconfig-* 命令管理全局 DNS 配置。全局 DNS 配置中定义的设置应用到所有 IdM DNS 服务器。

**配置 /etc/named.conf**

在每个 IdM DNS 服务器的 /etc/named.conf 文件中定义的配置具有最低优先级。它特定于每台服务器，必须手动编辑。

/etc/named.conf 文件通常仅用于指定到本地 DNS 缓存的 DNS 转发。其他选项通过使用 命令管

理上述区域特定和全局 DNS 配置。

您可以在多个级别上同时配置 DNS 选项。在这种情况下，具有最高优先级的配置优先于较低级别上定义的配置。

其它资源

- LDAP 中每服务配置 中的 配置的优先级顺序 部分

## 92.7. 主要 IDM DNS 区的配置属性

身份管理(IdM)创建一个具有特定默认配置的新区域，如刷新周期、传输设置或缓存设置。在 IdM DNS 区域属性 中，您可以使用以下选项之一查找默认区域配置的属性：

- 命令行界面(CLI)中的 dnszone-mod 命令。如需更多信息，请参阅在 IdM CLI 中编辑主 DNS 区的配置。
- IdM Web UI.如需更多信息，请参阅在 IdM Web UI 中编辑主 DNS 区的配置。
- 使用 ipadnszone 模块的 Ansible playbook。如需更多信息，请参阅在 IdM 中管理 DNS 区域。

除了设置区域的实际信息外，这些设置定义了 DNS 服务器如何处理 *权威启动* (SOA)记录条目以及它如何从 DNS 名称服务器更新其记录。

表 92.1. IdM DNS 区域属性

| 属性 | 命令行选项 | 描述 |
| --- | --- | --- |
| 权威名称服务器 | **--name-server** | 设置主 DNS 名称服务器的域名，也称为 SOA MNAME。<br><br>默认情况下，每个 IdM 服务器在 SOA MNAME 字段中公告其自身。因此，使用 **--name-server** 在 LDAP 中存储的值将被忽略。 |
| 管理员电子邮件地址 | **--admin-email** | 设置要用于区域管理员的电子邮件地址。这默认为主机上的 root 帐户。 |

| 属性 | 命令行选项 | 描述 |
| --- | --- | --- |
| SOA 串行 | **--serial** | 在 SOA 记录中设置序列号。请注意，IdM 会自动设置版本号，用户不应该修改它。 |
| SOA 刷新 | **--refresh** | 设置次要 DNS 服务器在从主 DNS 服务器请求更新前等待的间隔（以秒为单位）。 |
| SOA 重试 | **--retry** | 设置在重试失败的刷新操作前要等待的时间（以秒为单位）。 |
| SOA 过期 | **--expire** | 设置次要 DNS 服务器在结束操作尝试之前尝试执行刷新更新的时间（以秒为单位）。 |
| 最低 SOA | **--minimum** | 根据 RFC 2308，将生存时间(TTL)值（以秒为单位）设置为负缓存。 |
| SOA 生存时间 | **--ttl** | 为区域 apex 的记录设置 TTL（以秒为单位）。例如，在区域 **example.com** 中，配置了名称 **example.com** 下的所有记录(A、NS 或 SOA)，但其他域名（如 **test.example.com** ）受到了影响。 |
| 默认生存时间 | **--default-ttl** | 将默认时间设置为 live(TTL)，以秒为单位，为之前未设置单个 TTL 值的区域中的所有值提供负缓存。更改生效后，需要在所有 IdM DNS 服务器上重新启动 **named-pkcs11** 服务。 |
| BIND 更新策略 | **--update-policy** | 设置 DNS 区域中客户端允许的权限。 |
| 动态更新 | **--dynamic-update**=TRUE\|FALSE | 启用对客户端的 DNS 记录的动态更新。<br><br>请注意，如果设置为 false，IdM 客户端计算机将无法添加或更新其 IP 地址。 |
| 允许传输 | **--allow-transfer**=*string* | 提供允许传输给定区域的 IP 地址或网络名称列表，用分号(;)分隔。<br><br>默认情况下禁用区域传送。默认的 **--allow-transfer** 值为 **none**。 |
| 允许查询 | **--allow-query** | 提供允许发出 DNS 查询的 IP 地址或网络名称列表，用分号(;)分隔。 |
| 允许 PTR 同步 | **--allow-sync-ptr**=1\|0 | 设置区域的 A 或 AAAA 记录（转发记录）是否将自动与 PTR（反向）记录同步。 |
| 区域转发器 | **--forwarder**=*IP_address* | 指定为 DNS 区域特别配置的转发器。这与 IdM 域中使用的任何全局转发器分开。<br><br>要指定多个转发器，请多次使用 选项。 |

| 属性 | 命令行选项 | 描述 |
| --- | --- | --- |
| forward 策略 | **--forward-policy**=none\|only\|first | 指定 forward 策略。有关支持的策略的详情，请查看 IdM 中的 DNS 转发策略。 |

## 92.8. 在 IDM WEB UI 中编辑主 DNS 区域的配置

**按照以下流程，使用 IdM Web UI 编辑主身份管理(IdM) DNS 的配置属性。**

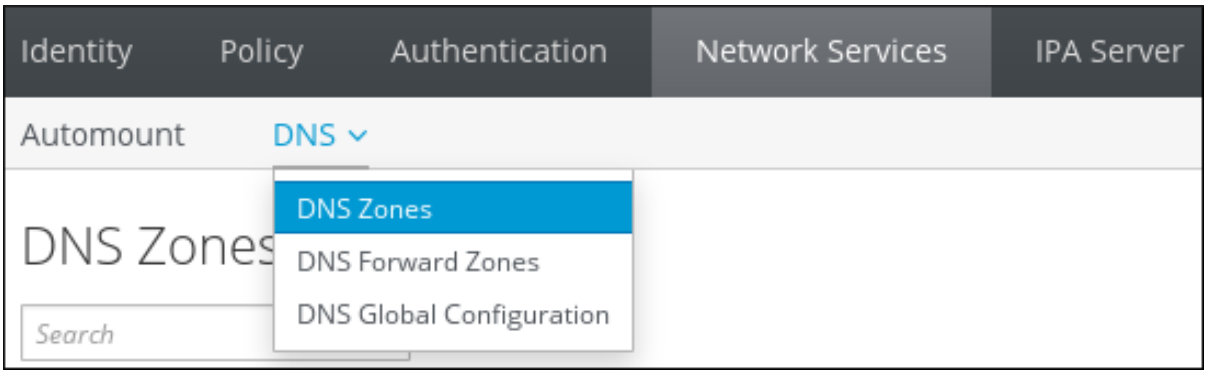
### 先决条件

- **以 IdM 管理员身份登录。**

### 流程

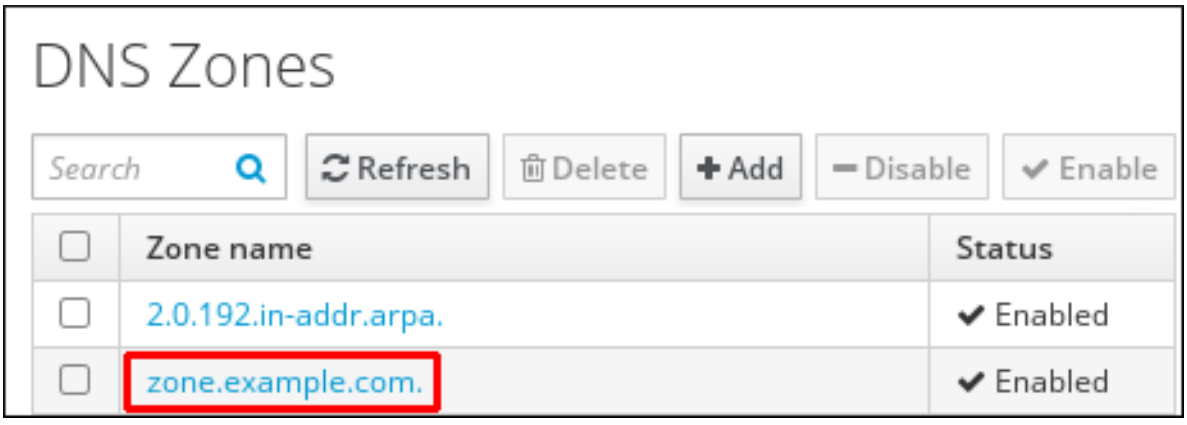
1. **在 IdM Web UI 中，点击 Network Services → DNS → DNS Zones。**

**图 92.4. DNS 主区管理**



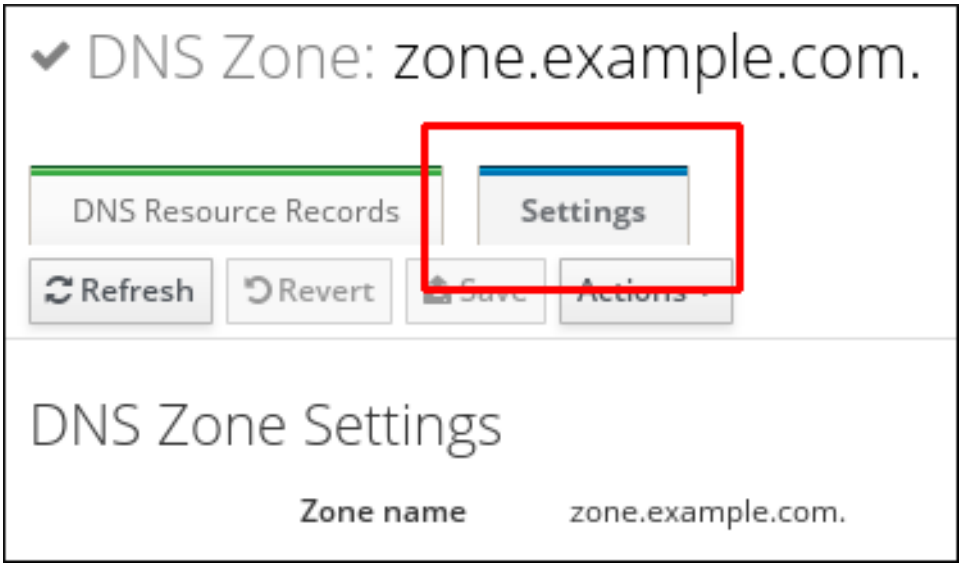

2. **在 DNS Zones 部分中，单击所有区域列表中的区域名称，以打开 DNS 区域页面。**

**图 92.5. 编辑主区域**



3. 单击 **Settings**。

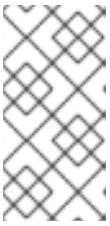
图 92.6. 主区域编辑页面中的 Settings 选项卡




4. **根据需要更改区域配置。**

   **有关可用设置的详情，请参考 IdM DNS 区域属性。**

5. 单击 **Save** 以确认新配置。



**注意**

**如果您要将所有区域的默认时间更改为 live(TTL)，在所有 IdM DNS 服务器上重新启动 named-pkcs11 服务，以使更改生效。所有其他设置将立即自动激活。**

## 92.9. 在 IDM CLI 中编辑主 DNS 区域的配置

**按照以下流程，使用身份管理(IdM)命令行界面(CLI)编辑主 DNS 区域的配置。**

**先决条件**

- **以 IdM 管理员身份登录。**

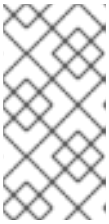
**流程**

- 要修改现有的主 DNS 区域，请使用 ipa dnszone-mod 命令。例如，在重试失败的刷新操作前要设置等待的时间为 1800 秒：

```
$ ipa dnszone-mod --retry 1800
```

有关可用设置及其相应 CLI 选项的更多信息，请参阅 IdM DNS 区域属性。

如果特定设置在您要修改的 DNS 区域条目中没有值，ipa dnszone-mod 命令会 添加该值。如果设置没有值，该命令将使用指定的值覆盖当前值。



注意

如果您要将所有区域的默认时间更改为 live(TTL)，在所有 IdM DNS 服务器上重新启动 named-pkcs11 服务，以使更改生效。所有其他设置将立即自动激活。

其它资源

- 请参阅 ipa dnszone-mod --help。

## 92.10. IDM 中的区域传送

在有集成 DNS 的身份管理(IdM)部署中，您可以使用 *zone transfers* 将所有资源记录从一个名称服务器复制到另一个名称服务器。名称服务器为其区域维护权威数据。如果您更改了对 *zone A* DNS 区域具有权威的 DNS 服务器上的区域，您必须在位于*zone A* 外的 IdM DNS 域中的其他名称服务器间分发更改。

重要

IdM 集成的 DNS 可以由不同的服务器同时写入。IdM 区域中的授权起始(SOA)序列号没有在单独的 IdM DNS 服务器间同步。因此，将您的 DNS 服务器配置为仅使用 to-be-transferred 区域中的一个特定的 DNS 服务器。这可防止由未同步的 SOA 序列号导致的区域传输失败。

IdM 支持根据 RFC 5936 (AXFR)和 RFC 1995( IXFR)标准进行区域传输。

其它资源

- 请参阅 在 IdM Web UI 中启用区域传送。
- 请参阅 在 IdM CLI 中启用区域传送。

## 92.11. 在 IDM WEB UI 中启用区传输

按照以下流程，使用 IdM Web UI 在身份管理(IdM)中启用区域传送。

### 先决条件

- 以 IdM 管理员身份登录。

### 流程

1. 在 IdM Web UI 中，点击 Network Services → DNS → DNS Zones。
2. 单击 Settings。
3. 在 Allow transfer 下，指定要将区域记录传输到的名称服务器。

图 92.7. 启用区传输


4. 单击 DNS 区域页面顶部的 Save，以确认新配置。

## 92.12. 在 IDM CLI 中启用区传输

按照以下流程，使用 IdM 命令行界面(CLI)在身份管理(IdM)中启用区域传送。

**先决条件**

- 以 IdM 管理员身份登录。
- 具有到辅助 DNS 服务器的 root 访问权限。

**流程**

- 要在 BIND 服务中启用区域传送，请输入 ipa dnszone-mod 命令，并指定位于 to-be-transferred 区域之外的名称服务器列表，以使用 --allow-transfer 选项将区域记录传输到其中。例如：

```
$ ipa dnszone-mod --allow-transfer=192.0.2.1;198.51.100.1;203.0.113.1
idm.example.com
```

验证步骤

1. SSH 到启用了区传输的 DNS 服务器之一：

```
$ ssh 192.0.2.1
```

2. 使用 dig 工具传输 IdM DNS 区域：

```
# dig @ipa-server zone_name AXFR
```

如果命令没有返回任何错误，则您已成功为 *zone_name* 启用区域传送。

## 92.13. 其它资源

- 请参阅 使用 Ansible playbook 来管理 IdM DNS 区域。

# 第 93 章 使用 ANSIBLE PLAYBOOK 管理 IDM DNS 区域

作为身份管理(IdM)管理员，您可以使用 ansible-freeipa 软件包中的 dnszone 模块来管理 IdM DNS 区域的工作方式。

- IdM 中支持哪些 DNS 区域类型
- 您可以在 IdM 中配置的 DNS 属性
- 如何使用 Ansible playbook 在 IdM DNS 中创建主区域
- 如何使用 Ansible playbook 确保存在带有多个变量的主 IdM DNS 区域
- 在提供 IP 地址时，如何使用 Ansible playbook 确保存在用于反向 DNS 查找的区域

**先决条件**

- DNS 服务安装在 IdM 服务器上。有关如何使用 Red Hat Ansible Engine 安装带有集成 DNS 的 IdM 服务器的更多信息，请参阅 使用 Ansible playbook 安装身份管理服务器。

## 93.1. 支持的 DNS 区类型

身份管理 (IdM) 支持两种类型的 DNS 区域：*primary* 和 *forward*区域。此处描述了这两种类型的区，包括 DNS 转发的示例场景。

**注意**

本指南对区域类型使用 BIND 术语，不同于用于 Microsoft Windows DNS 的术语。BIND 服务器中的 Primary zones 与 Microsoft Windows DNS 中的 *forward lookup zones* 和 *reverse lookup zones* 作用相同。BIND 中的转发区域的作用与 Microsoft Windows DNS *中的条件* 转发器相同。

**主 DNS 区域**

主 DNS 区域包含权威 DNS 数据，并且可以接受动态 DNS 更新。此行为等同于标准 BIND 配置中的 类型 master 设置。您可以使用 ipa dnszone-* 命令管理主区域。

根据标准 DNS 规则，每个主区域必须包含 授权起始 (SOA)和 名称服务器 (NS)记录。在创建 DNS 区域时，IdM 会自动生成这些记录，但您必须手动将 NS 记录复制到父区域，以创建正确委托。

根据标准 BIND 行为，查询服务器不具有权威的名称将转发到其他 DNS 服务器。这些 DNS 服务器（称为转发器）可能是也可能不是查询的权威。

> **例 93.1. DNS 转发的示例**
>
> IdM 服务器包含 test.example. 主区域。此区域包含 sub.test.example. 名称的 NS 委派记录。此外，使用 sub.text.example 子区域的 192.0.2.254 转发器 IP 地址配置了 test.example. 区域。
>
> 查询名称 不存在.test.example. 的客户端会收到 NXDomain 回答，并且不会发生转发，因为 IdM 服务器对此名称具有权威。
>
> 另一方面，查询 host1.sub.test.example. name 会转发到配置的转发器 192.0.2.254，因为 IdM 服务器对此名称没有权威。

转发 DNS 区域

从 IdM 的角度来看，转发 DNS 区域不包含任何权威数据。事实上，正向"区"通常仅包含两段信息：

- 域名
- 与域关联的 DNS 服务器的 IP 地址

所有对属于定义的域的名称的查询都转发到指定的 IP 地址。此行为等同于标准 BIND 配置中的 type forward 设置。您可以使用 ipa dnsforwardzone-* 命令管理转发区。

转发 DNS 区域在 IdM-Active Directory(AD)信任的上下文中特别有用。如果 IdM DNS 服务器对 idm.example.com 区域具有权威，并且 AD DNS 服务器对 ad.example.com 区域具有权威，则 ad.example.com 是 idm.example.com 主区的 DNS 转发区域。这意味着，当查询来自 IdM 客户端以获

取 somehost.ad.example.com 的 IP 地址时，查询将转发到 ad.example.com IdM DNS 转发区域中指定的 AD 域控制器。

### 93.2. 主要 IDM DNS 区的配置属性

身份管理(IdM)创建一个具有特定默认配置的新区域，如刷新周期、传输设置或缓存设置。在 IdM DNS 区域属性 中，您可以使用以下选项之一查找默认区域配置的属性：

- 命令行界面(CLI)中的 dnszone-mod 命令。如需更多信息，请参阅在 IdM CLI 中编辑主 DNS 区的配置。
- IdM Web UI.如需更多信息，请参阅在 IdM Web UI 中编辑主 DNS 区的配置。
- 使用 ipadnszone 模块的 Ansible playbook。如需更多信息，请参阅在 IdM 中管理 DNS 区域。

除了设置区域的实际信息外，这些设置定义了 DNS 服务器如何处理 *权威启动* (SOA)记录条目以及它如何从 DNS 名称服务器更新其记录。

表 93.1. IdM DNS 区域属性

| 属性 | ansible-freeipa 变量 | 描述 |
|---|---|---|
| 权威名称服务器 | **name_server** | 设置主 DNS 名称服务器的域名，也称为 SOA MNAME。<br><br>默认情况下，每个 IdM 服务器在 SOA MNAME 字段中公告其自身。因此，使用 **--name-server** 在 LDAP 中存储的值将被忽略。 |
| 管理员电子邮件地址 | **admin_email** | 设置要用于区域管理员的电子邮件地址。这默认为主机上的 root 帐户。 |
| SOA 串行 | **serial** | 在 SOA 记录中设置序列号。请注意，IdM 会自动设置版本号，用户不应该修改它。 |
| SOA 刷新 | **refresh** | 设置次要 DNS 服务器在从主 DNS 服务器请求更新前等待的间隔（以秒为单位）。 |
| SOA 重试 | **retry** | 设置在重试失败的刷新操作前要等待的时间（以秒为单位）。 |

| 属性 | ansible-freeipa 变量 | 描述 |
| --- | --- | --- |
| SOA 过期 | **expire** | 设置次要 DNS 服务器在结束操作尝试之前尝试执行刷新更新的时间（以秒为单位）。 |
| 最低 SOA | **最小值** | 根据 RFC 2308，将生存时间(TTL)值（以秒为单位）设置为负缓存。 |
| SOA 生存时间 | **ttl** | 为区域 apex 的记录设置 TTL（以秒为单位）。例如，在区域 **example.com** 中，配置了名称 **example.com** 下的所有记录(A、NS 或 SOA)，但其他域名（如 **test.example.com** ）受到了影响。 |
| 默认生存时间 | **default_ttl** | 将默认时间设置为 live(TTL)，以秒为单位，为之前未设置单个 TTL 值的区域中的所有值提供负缓存。更改生效后，需要在所有 IdM DNS 服务器上重新启动 **named-pkcs11** 服务。 |
| BIND 更新策略 | **update_policy** | 设置 DNS 区域中客户端允许的权限。 |
| 动态更新 | **dynamic_update**=TRUE\|FALSE | 启用对客户端的 DNS 记录的动态更新。<br><br>请注意，如果设置为 false，IdM 客户端计算机将无法添加或更新其 IP 地址。 |
| 允许传输 | **allow_transfer**=*string* | 提供允许传输给定区域的 IP 地址或网络名称列表，用分号(;)分隔。<br><br>默认情况下禁用区域传送。默认的 **allow_transfer** 值为 **none**。 |
| 允许查询 | **allow_query** | 提供允许发出 DNS 查询的 IP 地址或网络名称列表，用分号(;)分隔。 |
| 允许 PTR 同步 | **allow_sync_ptr**=1\|0 | 设置区域的 A 或 AAAA 记录（转发记录）是否将自动与 PTR（反向）记录同步。 |
| 区域转发器 | **forwarder**=*IP_address* | 指定为 DNS 区域特别配置的转发器。这与 IdM 域中使用的任何全局转发器分开。<br><br>要指定多个转发器，请多次使用 选项。 |
| forward 策略 | **forward_policy**=none\|only\|first | 指定 forward 策略。有关支持的策略的详情，请查看 IdM 中的 DNS 转发策略。 |

**其它资源**

- **请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnszone.md 文件。**

### 93.3. 使用 ANSIBLE 在 IDM DNS 中创建主区域

按照以下流程，使用 Ansible playbook 确保主 DNS 区域存在。在以下流程使用的示例中，您确保 zone.idm.example.com DNS 区域存在。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnszone 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 dnszone-present.yml Ansible playbook 文件的副本。例如：

```
$ cp dnszone-present.yml dnszone-present-copy.yml
```

4. 打开 dnszone-present-copy.yml 文件进行编辑。

5. 通过在 ipadnszone task 部分中设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 zone_name 变量设置为 zone.idm.example.com。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure zone is present.
    ipadnszone:
      ipaadmin_password: "{{ ipaadmin_password }}"
      zone_name: zone.idm.example.com
      state: present
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-present-copy.yml
```

**其它资源**

- 请参阅支持的 DNS 区域类型。

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnszone.md 文件。

- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnszone 目录中的 Ansible playbook 示例。

## 93.4. 使用 ANSIBLE PLAYBOOK 确保 IDM 中存在一个带有多个变量的主 DNS 区域

按照以下流程，使用 Ansible playbook 确保主 DNS 区域存在。在以下流程中使用的示例中，IdM 管理员确保存在 zone.idm.example.com DNS 区域。Ansible playbook 配置区域的多个参数。

### 先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

### 流程

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnszone 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 dnszone-all-params.yml Ansible playbook 文件的副本。例如：

```
$ cp dnszone-all-params.yml dnszone-all-params-copy.yml
```

4. 打开 dnszone-all-params-copy.yml 文件进行编辑。

5. 通过在 ipadnszone task 部分中设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 zone_name 变量设置为 zone.idm.example.com。
- 如果要允许正向和反向记录同步，这是 A 和 AAAA 记录与 PTR 记录的同步，请将 allow_sync_ptr 变量设置为 true。
- 将 dynamic_update 变量设置为 true，以启用 IdM 客户端计算机添加或更新其 IP 地址。
- 将 dnssec 变量设置为 true，以允许区域中的记录内联 DNSSEC 签名。
- 将 allow_transfer 变量设置为区域中次要名称服务器的 IP 地址。
- 将 allow_query 变量设置为允许发出查询的 IP 地址或网络。
- 将 forwarders 变量设置为全局转发器的 IP 地址。

- 将 serial 变量设置为 SOA 记录序列号。

- 为区域中的 DNS 记录定义 刷新、重试、过期 、 最小、ttl 和 default_ttl 值。

- 使用 nsec3param_rec 变量，为区域定义 NSEC3 PARAM 记录。

- 将 skip_overlap_check 变量设置为 true，从而强制创建 DNS，即使它与现有区域重叠。

- 将 skip_nameserver_check 设置为 true，从而强制 DNS 区域创建，即使名称服务器不可解析。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure zone is present.
    ipadnszone:
      ipaadmin_password: "{{ ipaadmin_password }}"
      zone_name: zone.idm.example.com
      allow_sync_ptr: true
      dynamic_update: true
      dnssec: true
      allow_transfer:
        - 1.1.1.1
        - 2.2.2.2
      allow_query:
        - 1.1.1.1
        - 2.2.2.2
      forwarders:
        - ip_address: 8.8.8.8
        - ip_address: 8.8.4.4
          port: 52
      serial: 1234
      refresh: 3600
      retry: 900
      expire: 1209600
      minimum: 3600
      ttl: 60
      default_ttl: 90
      name_server: server.idm.example.com.
```

```
    admin_email: admin.admin@idm.example.com
    nsec3param_rec: "1 7 100 0123456789abcdef"
    skip_overlap_check: true
    skip_nameserver_check: true
    state: present
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-all-params-copy.yml
```

**其它资源**

- 请参阅支持的 DNS 区域类型。
- 请参阅 主 IdM DNS 区域的配置属性。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnszone.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnszone 目录中的 Ansible playbook 示例。

## 93.5. 在给定 IP 地址时，使用 ANSIBLE PLAYBOOK 确保存在用于反向 DNS 查找的区域

按照以下流程，使用 Ansible playbook 确保反向 DNS 区域存在。在以下步骤中使用的示例中，IdM 管理员使用 IdM 主机的 IP 地址和前缀长度确保存在反向 DNS 查找区域。

使用 name_from_ip 变量提供 DNS 服务器的 IP 地址前缀长度，允许您控制区域名称。如果您不声明前缀长度，系统会查询 DNS 服务器以获取区，并根据 192.168.1.2 的 name_from_ip 值，查询可以返回以下 DNS 区域中的任何一个：

- 1.168.192.in-addr.arpa.

- 168.192.in-addr.arpa.
- 192.in-addr.arpa.

由于查询返回的区域可能不是您预期的区域，name_from_ip 只能与 state 选项设置为 present 一起使用，以防止意外删除区域。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnszone 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 dnszone-reverse-from-ip.yml Ansible playbook 文件的副本。例如：

```
$ cp dnszone-reverse-from-ip.yml dnszone-reverse-from-ip-copy.yml
```

4. 打开 dnszone-reverse-from-ip-copy.yml 文件进行编辑。

5. 通过在 ipadnszone task 部分中设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 name_from_ip 变量设置为 IdM 名称服务器的 IP，并提供其前缀长度。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure zone for reverse DNS lookup is present.
    ipadnszone:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name_from_ip: 192.168.1.2/24
      state: present
    register: result
  - name: Display inferred zone name.
    debug:
      msg: "Zone name: {{ result.dnszone.name }}"
```

playbook 创建一个区，用于从 192.168.1.2 IP 地址及其前缀长度 24 中反向 DNS 查找。接下来，playbook 显示生成的区域名称。

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-reverse-from-ip-copy.yml
```

**其它资源**

- 请参阅支持的 DNS 区域类型。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnszone.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnszone 目录中的 Ansible playbook 示例。

# 第 94 章 管理 IDM 中的 DNS 位置

要了解更多有关使用 IdM Web UI 和 IdM 命令行界面(CLI)管理身份管理(IdM) DNS 位置的信息，请参阅以下主题和流程：

- 基于 DNS 的服务发现
- DNS 位置的部署注意事项
- DNS 生存时间(TTL)
- 使用 IdM Web UI 创建 DNS 位置
- 使用 IdM CLI 创建 DNS 位置
- 使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置
- 使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置
- 将 IdM 客户端配置为使用同一位置的 IdM 服务器

## 94.1. 基于 DNS 的服务发现

在基于 DNS 的服务发现中，客户端使用 DNS 协议在提供特定服务的网络中查找服务器，如 LDAP 或 Kerberos。种典型的操作类型是允许客户端在最接近的网络基础架构中查找身份验证服务器，因为它们提供更高的吞吐量和较低的网络延迟，从而降低总体成本。

服务发现的主要优点是：

- 无需使用附近服务器的名称明确配置客户端。
-

DNS 服务器用作策略的中央提供程序。使用同一 DNS 服务器的客户端有权访问关于服务提供商及其首选顺序的相同策略。

在 Identity Management(IdM)域中，存在适用于 LDAP、Kerber os 和其他服务的 DNS 服务记录（SRV 记录）。例如，以下命令在 IdM DNS 域中查询 DNS 服务器以获取提供基于 TCP 的 Kerberos 服务的主机：

**例 94.1. 独立于 DNS 位置的结果**

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
0 100 88 idmserver-01.idm.example.com.
0 100 88 idmserver-02.idm.example.com.
```

输出包含以下信息：

- 0 （优先级）：目标主机的优先级.首选使用较低值。
- 100 (权重).为优先级相同的条目指定相对权重。如需更多信息，请参阅 RFC 2782 第 3 节。
- 88 （端口号）：服务的端口号。
- 提供服务的主机的规范名称。

在示例中，返回的两个主机名具有相同的优先级和权重。在本例中，客户端使用来自结果列表中的随机条目。

相反，当客户端配置为查询在 DNS 位置配置的 DNS 服务器时，输出会有所不同。对于分配到某个位置的 IdM 服务器，会返回定制值。在以下示例中，客户端被配置为在位置 germany 中查询 DNS 服务器：

**例 94.2. 基于 DNS 位置的结果**

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
_kerberos._tcp.germany._locations.idm.example.com.
0 100 88 idmserver-01.idm.example.com.
50 100 88 idmserver-02.idm.example.com.
```

IdM DNS 服务器自动返回一个 DNS 别名(CNAME)，指向一个 DNS 位置特定的 SRV 记录（首选本地服务器）。此 CNAME 记录显示在输出的第一行中。在示例中，主机 idmserver-01.idm.example.com 具有最低的优先级值，因此是首选的。idmserver-02.idm.example.com 具有更高的优先级，因此仅在首选主机不可用的情况下用作备份。

## 94.2. DNS 位置的部署注意事项

使用集成 DNS 时，身份管理(IdM)可以生成特定于位置的服务(SRV)记录。因为每个 IdM DNS 服务器都会生成特定于位置的 SRV 记录，所以您必须在每个 DNS 位置至少安装一个 IdM DNS 服务器。

客户端与 DNS 位置的关联仅由客户端收到的 DNS 记录定义。因此，如果客户端执行 DNS 服务发现，从 IdM DNS 服务器解析特定于位置的记录，您可以将 IdM DNS 服务器与非 IdM DNS 使用者服务器合并，并递归器。

在大多数使用混合 IdM 和非 IdM DNS 服务的部署中，DNS 递归器都通过使用往返时间指标自动选择最接近的 IdM DNS 服务器。通常，这可确保使用非 IdM DNS 服务器的客户端正在获取最接近的 DNS 位置的记录，从而使用最佳 IdM 服务器集。

## 94.3. DNS 生存时间(TTL)

客户端可以在区域配置中设置的一段时间内缓存 DNS 资源记录。由于这种缓存，客户端可能无法接收更改，直到生存时间(TTL)值过期为止。Identity Management(IdM)中的默认 TTL 值为 1 天。

如果您的客户端计算机在站点间漫游，您应该调整 IdM DNS 区的 TTL 值。将值设为比客户端在站点之间 roam 需要的时间值低。这样可确保客户端上缓存的 DNS 条目在重新连接到另一个站点之前过期，从而查询 DNS 服务器刷新特定于位置的 SRV 记录。

**其它资源**

- 请参阅 主 IdM DNS 区域的配置属性。

## 94.4. 使用 IDM WEB UI 创建 DNS 位置

您可以使用 DNS 位置来加快身份管理(IdM)客户端和服务器之间的通信速度。按照以下流程，使用 IdM Web UI 创建 DNS 位置。

**先决条件**

**先决条件**

- 您的 IdM 部署已经集成了 DNS。
- 您有在 IdM 中创建 DNS 位置的权限。例如，您以 IdM admin 身份登录。

**流程**

1. 打开 IPA Server 选项卡。
2. 选择 Topology 子选项卡。
3. 单击导航栏中的 IPA Locations。
4. 单击位置列表顶部的 Add。
5. 填写位置名称。
6. 单击 添加 按钮以保存位置。
7. 可选：重复添加更多位置的步骤。

**其它资源**

- 请参阅 使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置。
- 请参阅 使用 Ansible 来确保 IdM 位置存在。

### 94.5. 使用 IDM CLI 创建 DNS 位置

您可以使用 DNS 位置来加快身份管理(IdM)客户端和服务器之间的通信速度。按照以下流程，使用 IdM 命令行界面(CLI)中的 ipa location-add 命令创建 DNS 位置。

**先决条件**

- 您的 IdM 部署已经集成了 DNS。
- 您有在 IdM 中创建 DNS 位置的权限。例如，您以 IdM admin 身份登录。

**流程**

1. 例如，要创建新位置 germany，请输入：

```
$ ipa location-add germany
---------------------------
Added IPA location "germany"
---------------------------
  Location name: germany
```

2. 可选：重复添加更多位置的步骤。

**其它资源**

- 请参阅 使用 IdM CLI 将 IdM 服务器分配给 DNS 位置。
- 请参阅 使用 Ansible 来确保 IdM 位置存在。

### 94.6. 使用 IDM WEB UI 将 IDM 服务器分配给 DNS 位置

您可以使用 Identity Management(IdM)DNS 位置提高 IdM 客户端和服务器之间的通信速度。按照以下流程，使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置。

**先决条件**

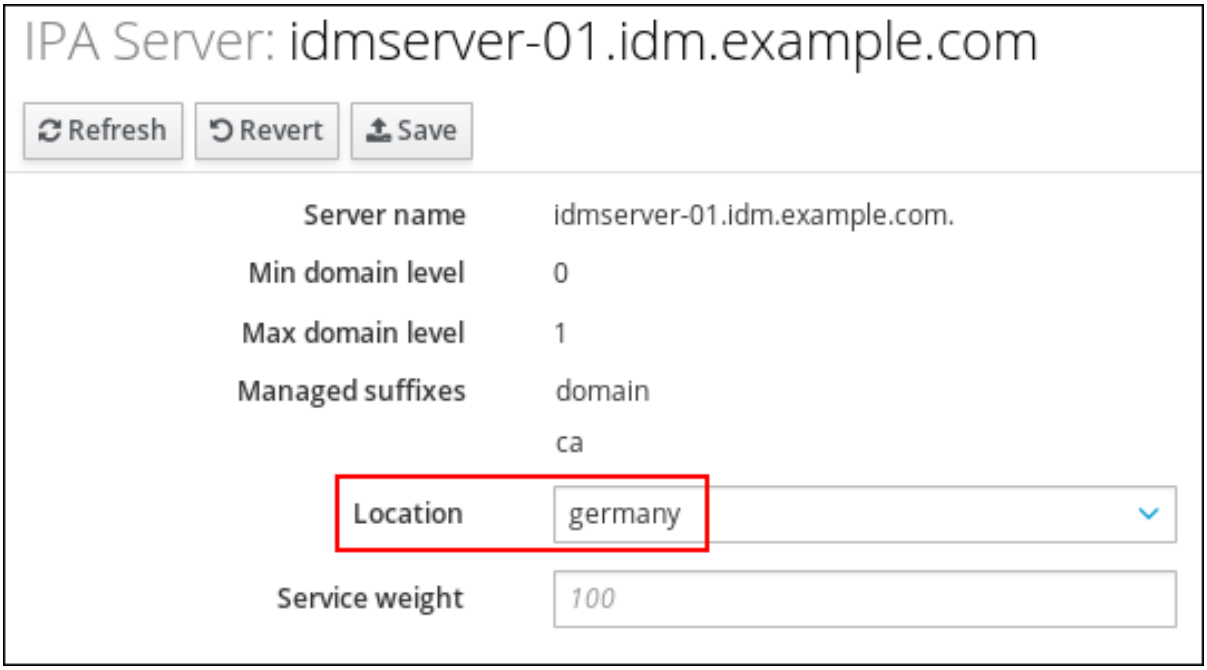
- 您的 IdM 部署已经集成了 DNS。
- 您以有权将服务器分配到 DNS 位置的用户身份登录，例如 IdM admin 用户。

- 您对要为其分配 DNS 位置的主机具有 root 访问权限。
- 您已 创建了您要将服务器分配到的 IdM DNS 位置。

**流程**

1. 打开 IPA Server 选项卡。
2. 选择 Topology 子选项卡。
3. 单击导航中的 IPA Servers。
4. 单击 IdM 服务器名称。
5. 选择 DNS 位置，并选择性地设置服务权重：

**图 94.1. 将服务器分配到 DNS 位置**




6. 点 Save。

7. 在您在前面的步骤中指定的主机的命令行界面(CLI)中，重启 named-pkcs11 服务：

```
[root@idmserver-01 ~]# systemctl restart named-pkcs11
```

8. 可选：重复为其他 IdM 服务器分配 DNS 位置的步骤。

**其它资源**

- 请参阅 配置 IdM 客户端以在同一位置上使用 IdM 服务器。

## 94.7. 使用 IDM CLI 将 IDM 服务器分配给 DNS 位置

您可以使用 Identity Management(IdM)DNS 位置提高 IdM 客户端和服务器之间的通信速度。按照以下流程，使用 IdM 命令行界面(CLI)将 IdM 服务器分配给 DNS 位置。

**先决条件**

- 您的 IdM 部署已经集成了 DNS。
- 您以有权将服务器分配到 DNS 位置的用户身份登录，例如 IdM admin 用户。
- 您对要为其分配 DNS 位置的主机具有 root 访问权限。
- 您已 创建了您要将服务器分配到的 IdM DNS 位置。

**流程**

1. 可选：列出所有配置的 DNS 位置：

```
[root@server ~]# ipa location-find
----------------------
2 IPA locations matched
----------------------
Location name: australia
Location name: germany
```

```
----------------------------
Number of entries returned: 2
----------------------------
```

2. 将服务器分配到 DNS 位置。例如，要将位置 germany 分配给服务器 idmserver-01.idm.example.com，请运行：

```
# ipa server-mod idmserver-01.idm.example.com --location=germany
ipa: WARNING: Service named-pkcs11.service requires restart on IPA server idmserver-01.idm.example.com to apply configuration changes.
-----------------------------------------------
Modified IPA server "idmserver-01.idm.example.com"
-----------------------------------------------
Servername: idmserver-01.idm.example.com
Min domain level: 0
Max domain level: 1
Location: germany
Enabled server roles: DNS server, NTP server
```

3. 在您在前面的步骤中指定的主机上重启 named-pkcs11 服务：

```
# systemctl restart named-pkcs11
```

4. 可选：重复为其他 IdM 服务器分配 DNS 位置的步骤。

**其它资源**

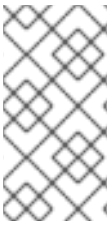
- 请参阅 配置 IdM 客户端以在同一位置上使用 IdM 服务器。

## 94.8. 将 IDM 客户端配置为使用同一位置的 IDM 服务器

身份管理(IdM)服务器被分配给 DNS 位置，如 使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置 中所述。现在，您可以将客户端配置为使用与 IdM 服务器位于同一位置的 DNS 服务器：

- 如果 DHCP 服务器为客户端分配 DNS 服务器 IP 地址，请配置 DHCP 服务。有关在 DHCP 服务中分配 DNS 服务器的详情，请查看 DHCP 服务文档。

- 如果您的客户端没有从 DHCP 服务器接收 DNS 服务器 IP 地址，请在客户端的网络配置中手动设置 IP。有关在 Red Hat Enterprise Linux 上配置网络的详情，请查看 *Red Hat Enterprise Linux 网络指南中* 的配置网络连接 设置 部分。



**注意**

如果您将客户端配置为使用分配给不同位置的 DNS 服务器，客户端会联系两个位置的 IdM 服务器。

**例 94.3. 根据客户端的位置的不同名称服务器条目**

以下示例显示了位于不同位置的客户端的 /etc/resolv.conf 文件中的不同名称服务器条目：

布拉格 中的客户端：

```
nameserver 10.10.0.1
nameserver 10.10.0.2
```

智利中的客户端：

```
nameserver 10.50.0.1
nameserver 10.50.0.3
```

Oslo 中的客户端：

```
nameserver 10.30.0.1
```

布林中的客户端：

```
nameserver 10.30.0.1
```

如果每个 DNS 服务器都被分配给 IdM 中的一个位置，客户端将使用其位置中的 IdM 服务器。

## 94.9. 其它资源

- 请参阅 在 IdM 中使用 Ansible 来管理 DNS 位置。

# 第 95 章 使用 ANSIBLE 管理 IDM 中的 DNS 位置

作为身份管理(IdM)管理员，您可以使用 ansible-freeipa 软件包中提供 的位置模块管理 IdM DNS 位置。

- 基于 DNS 的服务发现
- DNS 位置的部署注意事项
- DNS 生存时间(TTL)
- 使用 Ansible 确保存在 IdM 位置
- 使用 Ansible 确保缺少 IdM 位置

## 95.1. 基于 DNS 的服务发现

在基于 DNS 的服务发现中，客户端使用 DNS 协议在提供特定服务的网络中查找服务器，如 LDAP 或 Kerberos。种典型的操作类型是允许客户端在最接近的网络基础架构中查找身份验证服务器，因为它们提供更高的吞吐量和较低的网络延迟，从而降低总体成本。

服务发现的主要优点是：

- 无需使用附近服务器的名称明确配置客户端。
- DNS 服务器用作策略的中央提供程序。使用同一 DNS 服务器的客户端有权访问关于服务提供商及其首选顺序的相同策略。

在 Identity Management(IdM)域中，存在适用于 LDAP、Kerber os 和其他服务的 DNS 服务记录（SRV 记录）。例如，以下命令在 IdM DNS 域中查询 DNS 服务器以获取提供基于 TCP 的 Kerberos 服务的主机：

**例 95.1. 独立于 DNS 位置的结果**

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
0 100 88 idmserver-01.idm.example.com.
0 100 88 idmserver-02.idm.example.com.
```

输出包含以下信息：

- 0 （优先级）：目标主机的优先级.首选使用较低值。
- 100 (权重).为优先级相同的条目指定相对权重。如需更多信息，请参阅 RFC 2782 第 3 节。
- 88 （端口号）：服务的端口号。
- 提供服务的主机的规范名称。

在示例中，返回的两个主机名具有相同的优先级和权重。在本例中，客户端使用来自结果列表中的随机条目。

相反，当客户端配置为查询在 DNS 位置配置的 DNS 服务器时，输出会有所不同。对于分配到某个位置的 IdM 服务器，会返回定制值。在以下示例中，客户端被配置为在位置 germany 中查询 DNS 服务器：

例 95.2. 基于 DNS 位置的结果

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
_kerberos._tcp.germany._locations.idm.example.com.
0 100 88 idmserver-01.idm.example.com.
50 100 88 idmserver-02.idm.example.com.
```

IdM DNS 服务器自动返回一个 DNS 别名(CNAME)，指向一个 DNS 位置特定的 SRV 记录（首选本地服务器）。此 CNAME 记录显示在输出的第一行中。在示例中，主机 idmserver-01.idm.example.com 具有最低的优先级值，因此是首选的。idmserver-02.idm.example.com 具有更高的优先级，因此仅在首选主机不可用的情况下用作备份。

## 95.2. DNS 位置的部署注意事项

使用集成 DNS 时，身份管理(IdM)可以生成特定于位置的服务(SRV)记录。因为每个 IdM DNS 服务器都会生成特定于位置的 SRV 记录，所以您必须在每个 DNS 位置至少安装一个 IdM DNS 服务器。

客户端与 DNS 位置的关联仅由客户端收到的 DNS 记录定义。因此，如果客户端执行 DNS 服务发现，从 IdM DNS 服务器解析特定于位置的记录，您可以将 IdM DNS 服务器与非 IdM DNS 使用者服务器合并，并递归器。

在大多数使用混合 IdM 和非 IdM DNS 服务的部署中，DNS 递归器都通过使用往返时间指标自动选择最接近的 IdM DNS 服务器。通常，这可确保使用非 IdM DNS 服务器的客户端正在获取最接近的 DNS 位置的记录，从而使用最佳 IdM 服务器集。

### 95.3. DNS 生存时间(TTL)

客户端可以在区域配置中设置的一段时间内缓存 DNS 资源记录。由于这种缓存，客户端可能无法接收更改，直到生存时间(TTL)值过期为止。Identity Management(IdM)中的默认 TTL 值为 1 天。

如果您的客户端计算机在站点间漫游，您应该调整 IdM DNS 区的 TTL 值。将值设为比客户端在站点之间 roam 需要的时间值低。这样可确保客户端上缓存的 DNS 条目在重新连接到另一个站点之前过期，从而查询 DNS 服务器刷新特定于位置的 SRV 记录。

**其它资源**

- 请参阅 主 IdM DNS 区域的配置属性。

### 95.4. 使用 ANSIBLE 确保存在 IDM 位置

作为身份管理系统管理员(IdM)，您可以配置 IdM DNS 位置，以允许客户端在最接近的网络基础架构中查找身份验证服务器。

以下流程描述了如何使用 Ansible playbook 来确保 IdM 中存在 DNS 位置。这个示例描述了如何确保 IdM 中存在 germany DNS 位置。因此，您可以将特定的 IdM 服务器分配给这个位置，以便本地 IdM 客户端可以使用它们来缩短服务器响应时间。

**先决条件**

- 您知道 IdM 管理员密码。

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您了解 DNS 位置的部署注意事项。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 制作 /usr/share/doc/ansible-freeipa/playbooks/location/ 目录中 location-present.yml 文件的一个副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/location/location-present.yml location-present-copy.yml
```

3. 打开 location-present-copy.yml Ansible playbook 文件以进行编辑。
4. 通过在 ipalocation task 部分中设置以下变量来调整文件：
   - 调整任务的 name，使其与您的用例对应。

- 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
- 将 name 变量设置为位置的名称。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: location present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "germany" location is present
    ipalocation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: germany
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory location-present-copy.yml
```

**其它资源**

- 请参阅 使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置 或 使用 IdM CLI 将 IdM 服务器分配给 DNS 位置。

### 95.5. 使用 ANSIBLE 确保缺少 IDM 位置

作为身份管理系统管理员(IdM)，您可以配置 IdM DNS 位置，以允许客户端在最接近的网络基础架构中查找身份验证服务器。

以下流程描述了如何使用 Ansible playbook 来确保 IdM 中没有 DNS 位置。这个示例描述了如何确保 IdM 中没有 germany DNS 位置。因此，您无法将特定的 IdM 服务器分配给这个位置，本地 IdM 客户端

无法使用它们。

**先决条件**

- 您知道 IdM 管理员密码。
- 没有 IdM 服务器被分配给 germany DNS 位置。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 示例假定您已 创建并配置了 ~/*MyPlaybooks*/ 目录，来作为存储示例 playbook 副本的中心位置。

**流程**

1. 进入 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 为 /usr/share/doc/ansible-freeipa/playbooks/location/ 目录中的location-absent.yml 文件制作一个副本：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/location/location-absent.yml location-absent-copy.yml
```

3. 打开 location-absent-copy.yml Ansible playbook 文件以进行编辑。

4. 通过在 ipalocation task 部分中设置以下变量来调整文件：

   - 调整任务的 name，使其与您的用例对应。
   - 将 ipaadmin_password 变量设置为 IdM 管理员的密码。
   - 将 name 变量设置为 DNS 位置的名称。
   - 确保 state 变量设置为 absent。

   这是当前示例修改的 Ansible playbook 文件：

```
---
- name: location absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "germany" location is absent
    ipalocation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: germany
      state: absent
```

5. 保存该文件。

6. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory location-absent-copy.yml
```

### 95.6. 其它资源

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-location.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/location 目录中的 Ansible playbook 示例。

# 第 96 章 在 IDM 中管理 DNS 转发

按照以下流程，在身份管理(IdM) Web UI、IdM CLI 中以及使用 Ansible 来配置 DNS 全局转发器和 DNS 转发区域：

- IdM DNS 服务器的两个角色
- IdM 中的 DNS 转发策略
- 在 IdM Web UI 中添加全局转发器
- 在 CLI 中添加全局转发器
- 在 IdM Web UI 中添加 DNS 转发区域
- 在 CLI 中添加 DNS 转发区域
- 使用 Ansible 在 IdM 中建立 DNS 全局转发器
- 使用 Ansible 确保 IdM 中存在 DNS 全局转发器
- 使用 Ansible 确保 IdM 中没有 DNS 全局转发器
- 使用 Ansible 确保 DNS 全局转发器在 IdM 中被禁用
- 使用 Ansible 确保 IdM 中存在 DNS 转发区域
- 使用 Ansible 确保 DNS 转发区域 在 IdM 中有多个转发器

- 使用 Ansible 确保 IdM 中 DNS Forward 区域被禁用
- 使用 Ansible 确保 IdM 中没有 DNS 转发区域

## 96.1. IDM DNS 服务器的两个角色

DNS 转发会影响 DNS 服务如何应答 DNS 查询。默认情况下，集成了 IdM 的 Berkeley Internet Name Domain (BIND) 作为一个 *authoritative* 和一个 *recursive* DNS 服务器：

权威 DNS 服务器

当 DNS 客户端查询属于 IdM 服务器具有权威的 DNS 区域的名称时，BIND 回复包含在配置区域中的数据。权威数据总是优先于任何其他数据。

递归 DNS 服务器

当 DNS 客户端查询 IdM 服务器不是权威的名称时，BIND 会尝试使用其他 DNS 服务器解析查询。如果未定义转发器，BIND 会询问 Internet 上的根服务器，并使用递归解析算法回答 DNS 查询。

在某些情况下，不需要让 BIND 直接联系其他 DNS 服务器，并根据 Internet 上可用的数据执行递归。您可以将 BIND 配置为使用另一个 DNS 服务器（*转发器*）来解析查询。

当您将 BIND 配置为使用转发器时，查询和答案将在 IdM 服务器和转发器之间来回转发，IdM 服务器充当非权威数据的 DNS 缓存。

## 96.2. IDM 中的 DNS 转发策略

IdM 支持 第一个 且 唯一的 标准 BIND 转发策略，以及任何 IdM 特定的转发策略。

首先转发 *（默认）*

IdM BIND 服务将 DNS 查询转发到配置的转发器。如果因为服务器错误或超时查询失败，BIND 会使用 Internet 上的服务器回退到递归解析。forward first 策略是默认策略，它适合优化 DNS 流量。

仅转发

IdM BIND 服务将 DNS 查询转发到配置的转发器。如果因为服务器错误或超时查询失败，BIND 会向客户端返回错误。建议在带有分割 DNS 配置的环境中使用 forward only 策略。

**none** *（转发禁用）*

DNS 查询不会通过 none 转发策略转发。禁用转发仅作为全局转发配置的特定区覆盖。这个选项等同于在 BIND 配置中指定空转发器列表。

**注意**

您不能使用转发将 IdM 中的数据与其他 DNS 服务器的数据组合。您只能在 IdM DNS 中转发主区的查询。

默认情况下，如果查询的 DNS 名称属于 IdM 服务器具有权威的区域，BIND 服务不会将查询转发到另一服务器。在这种情况下，如果无法在 IdM 数据库中找到查询的 DNS 名称，则会返回 NXDOMAIN 回答。未使用转发。

**例 96.1. 场景示例**

IdM 服务器对 test.example 具有权威。DNS 区域.BIND 配置为将查询转发到 IP 地址 192.0.2.254 的 DNS 服务器。

客户端发送对 不存在.test.example的查询时。DNS 名称，BIND 检测到 IdM 服务器对 test.example. 区域具有权威，并且不会将查询转发到 192.0.2.254. 服务器。因此，DNS 客户端会收到 NXDomain 错误消息，通知用户查询的域不存在。

## 96.3. 在 IDM WEB UI 中添加全局转发器

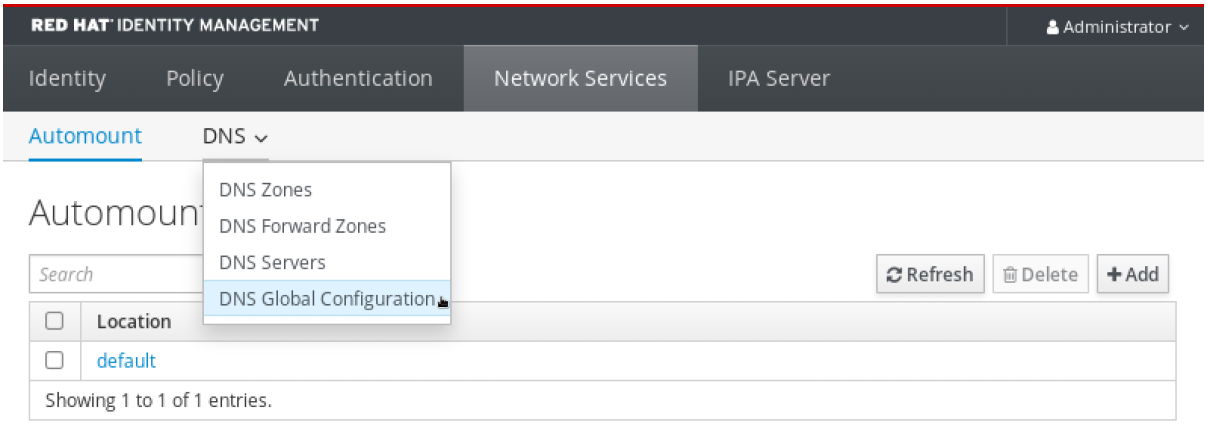
按照以下流程在身份管理(IdM) Web UI 中添加全局 DNS 转发器。

**先决条件**

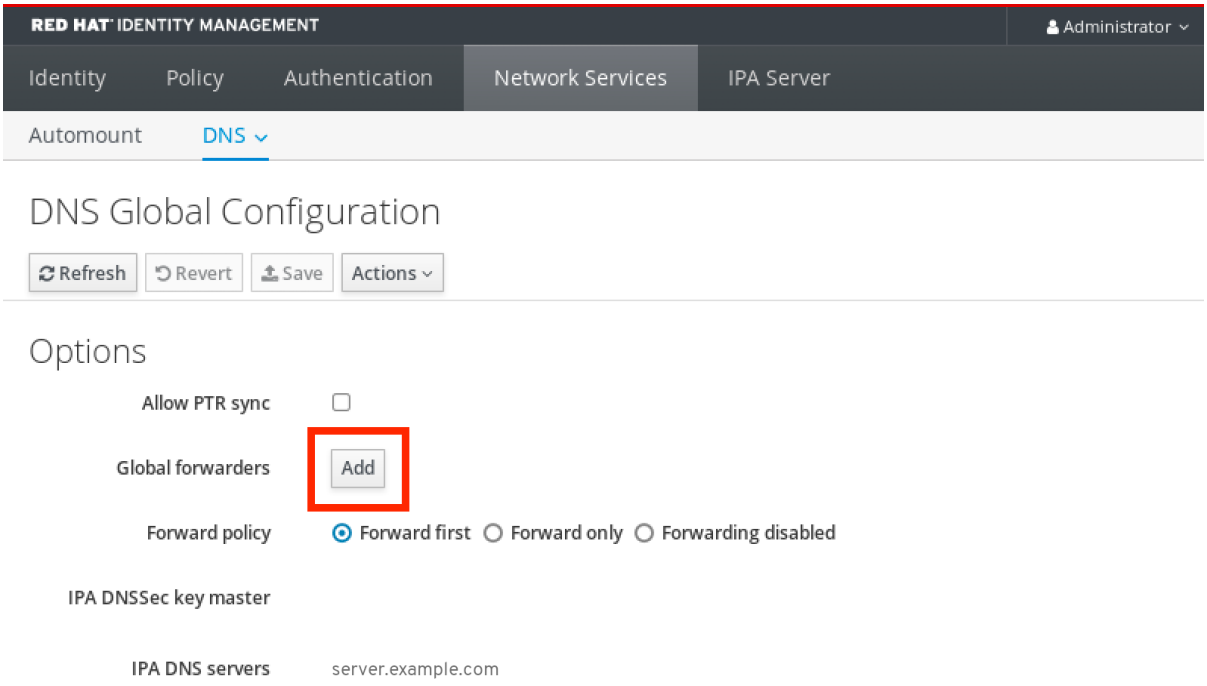
- 以 IdM 管理员身份登录到 IdM WebUI。
- 您知道要将查询转发到的 DNS 服务器的 Internet 协议(IP)地址。

**流程**

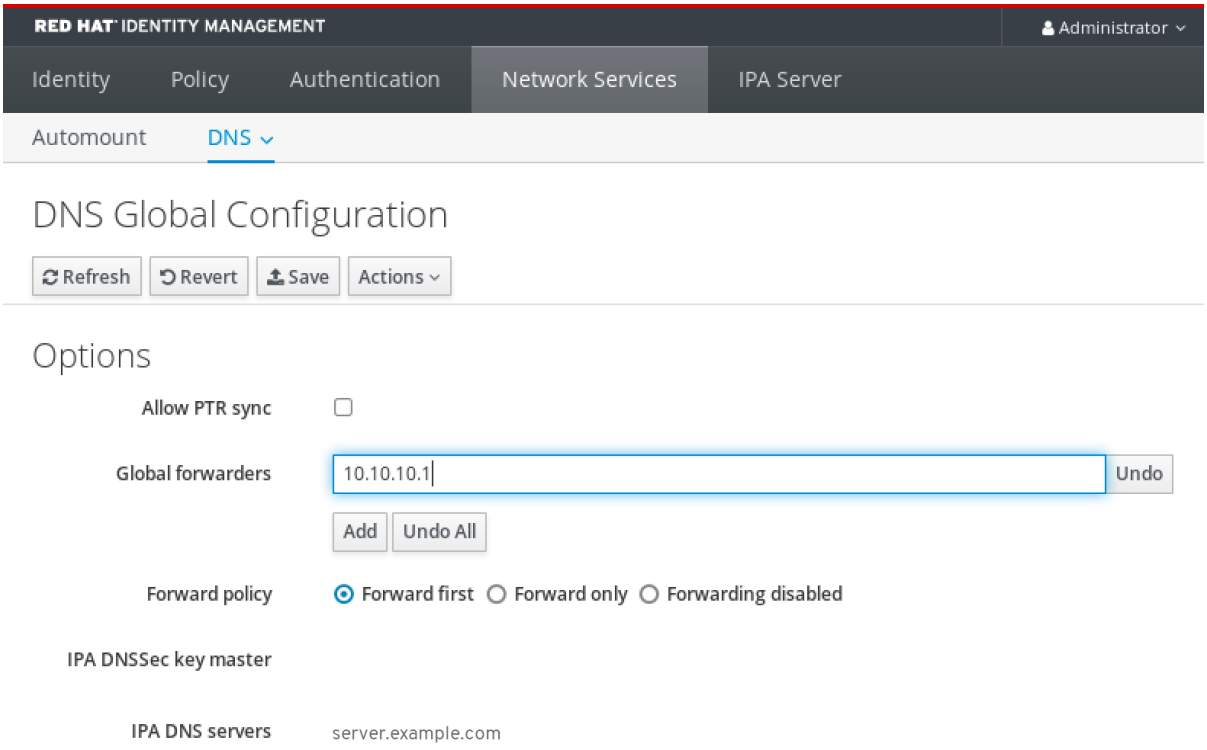
1. 在 IdM Web UI 中，选择 Network Services → DNS Global Configuration → DNS。



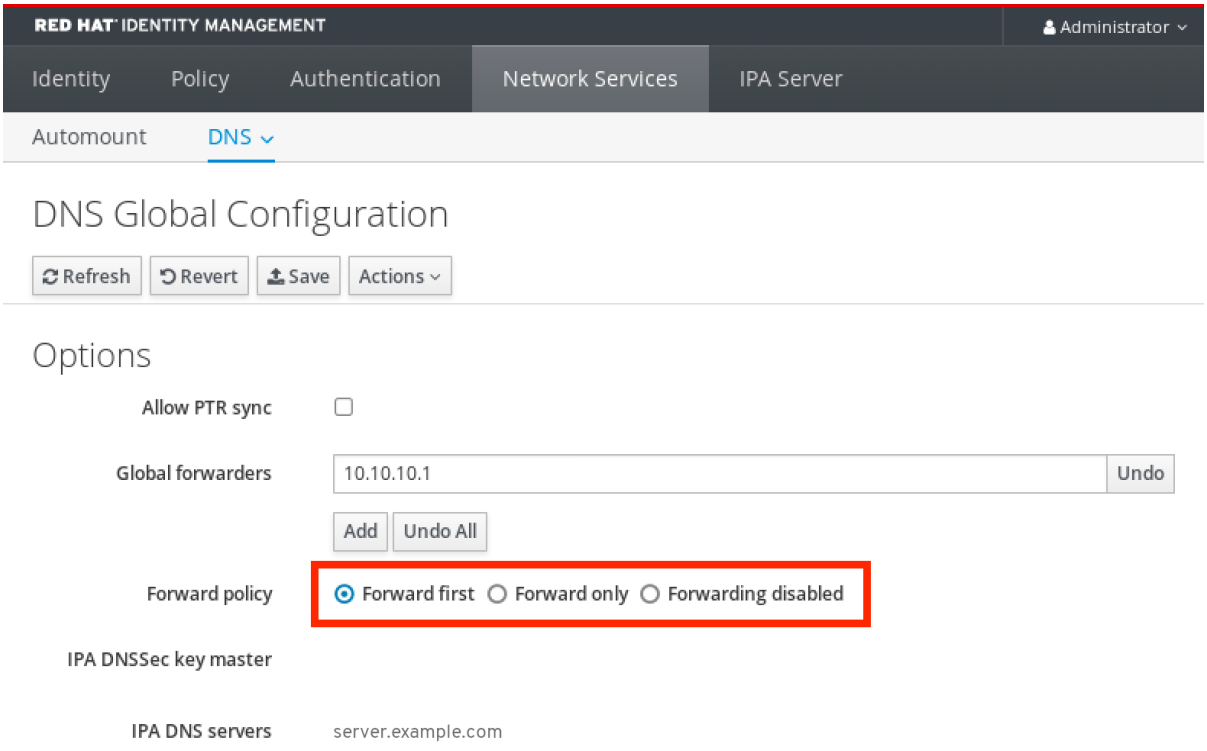

2. **在 DNS Global Configuration 部分中，单击 Add。**




3. **指定将接收转发 DNS 查询的 DNS 服务器的 IP 地址。**



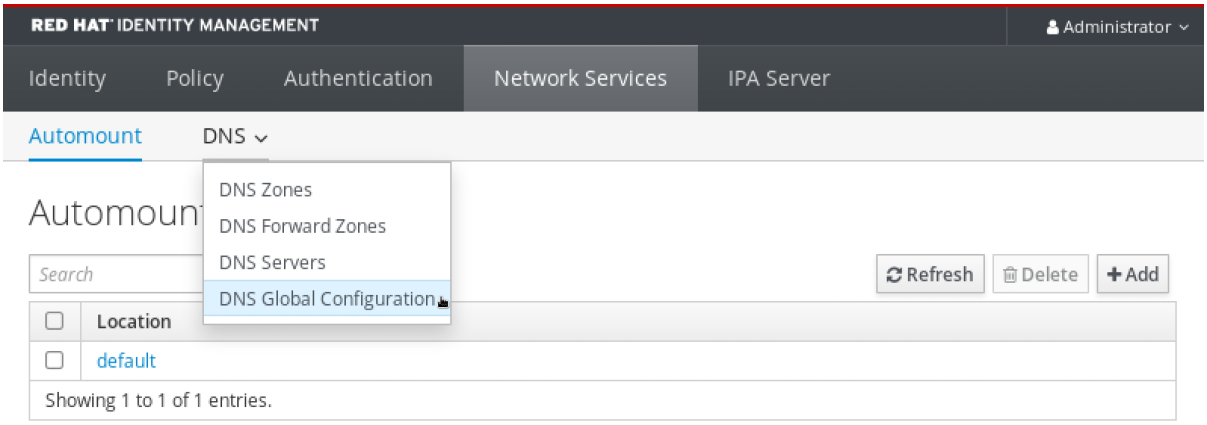

4. 选择 **Forward** 策略。



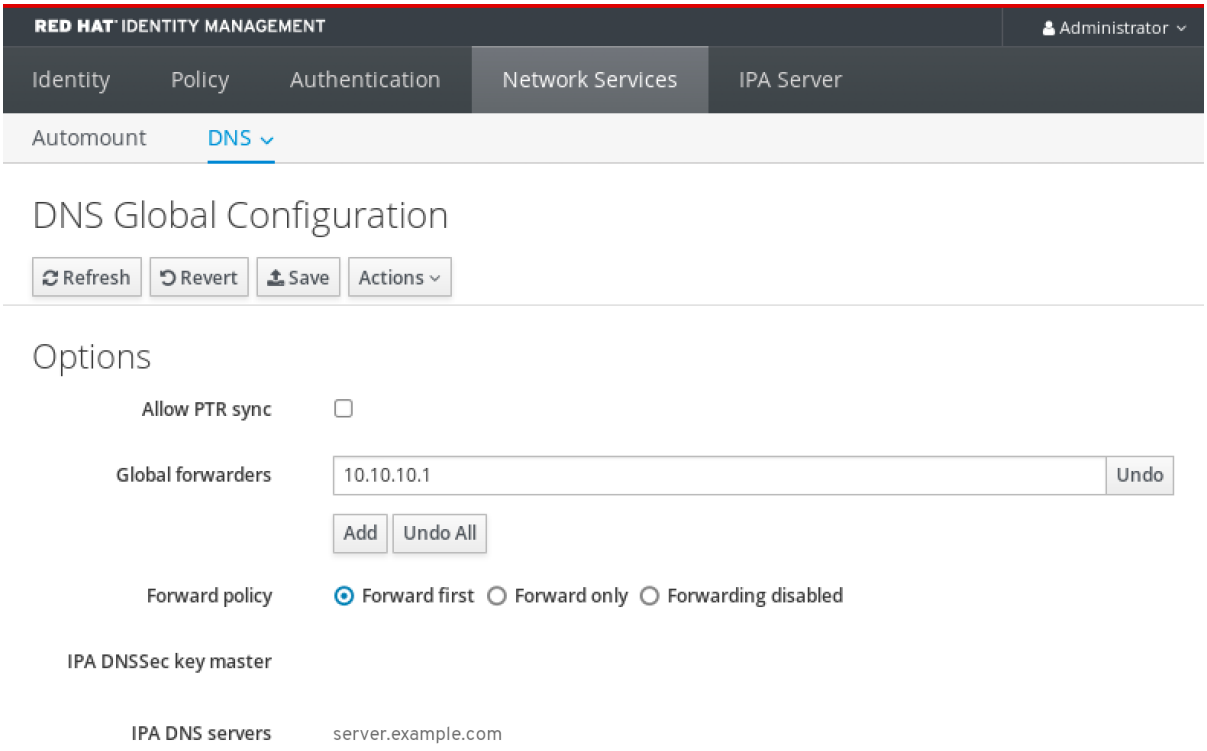

5. 单击窗口顶部的 **Save**。

**验证步骤**

1. 选择 **Network Services → DNS Global Configuration → DNS**。




2. 验证 **IdM Web UI 中是否存在并启用了带有您指定的 forward 策略的全局转发器。**




## 96.4. 在 CLI 中添加全局转发器

**按照以下流程，使用命令行界面(CLI)添加全局 DNS 转发器。**

**先决条件**

- **以 IdM 管理员身份登录。**

- 您知道要将查询转发到的 DNS 服务器的 Internet 协议(IP)地址。

流程

- 使用 ipa dnsconfig-mod 命令添加新的全局转发器。使用 --forwarder 选项指定 DNS 转发器的 IP 地址。

```
[user@server ~]$ ipa dnsconfig-mod --forwarder=10.10.0.1
Server will check DNS forwarder(s).
This may take some time, please wait ...
  Global forwarders: 10.10.0.1
  IPA DNS servers: server.example.com
```

验证步骤

- 使用 dnsconfig-show 命令显示全局转发器。

```
[user@server ~]$ ipa dnsconfig-show
  Global forwarders: 10.10.0.1
  IPA DNS servers: server.example.com
```

### 96.5. 在 IDM WEB UI 中添加 DNS 转发区域

按照以下流程在身份管理(IdM) Web UI 中添加 DNS 转发区域。

重要

除非绝对需要，否则请不要使用转发区域。转发区域不是标准解决方案，使用它们可能会导致意外和有问题的行为。如果您必须使用 forward zone，限制使用它们覆盖全局转发配置。

在创建新 DNS 区域时，红帽建议您始终使用名称服务器(NS)记录和避免转发区域，始终使用标准 DNS 委派。在大多数情况下，使用全局转发器足够了，并且转发区不需要。

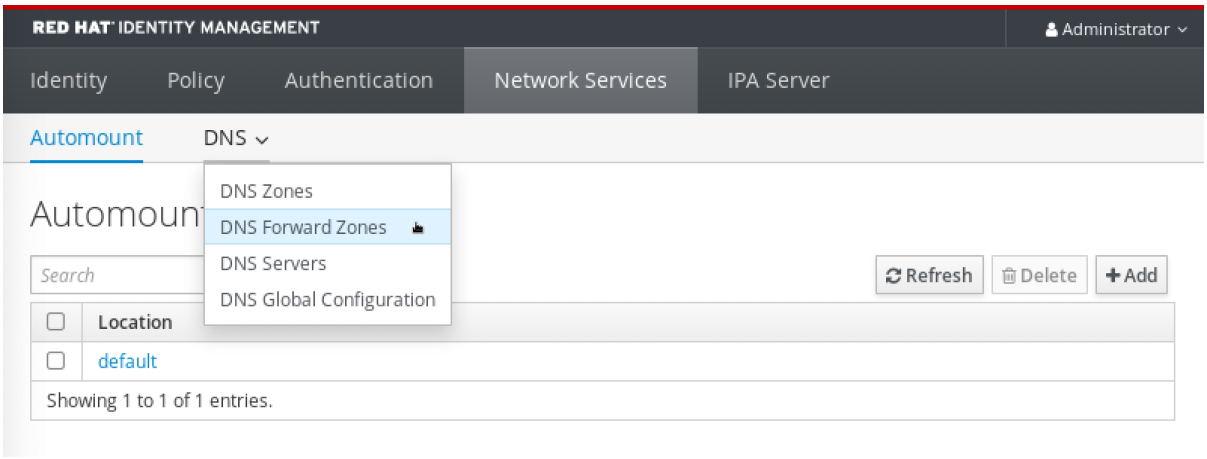
先决条件

- 以 IdM 管理员身份登录到 IdM WebUI。

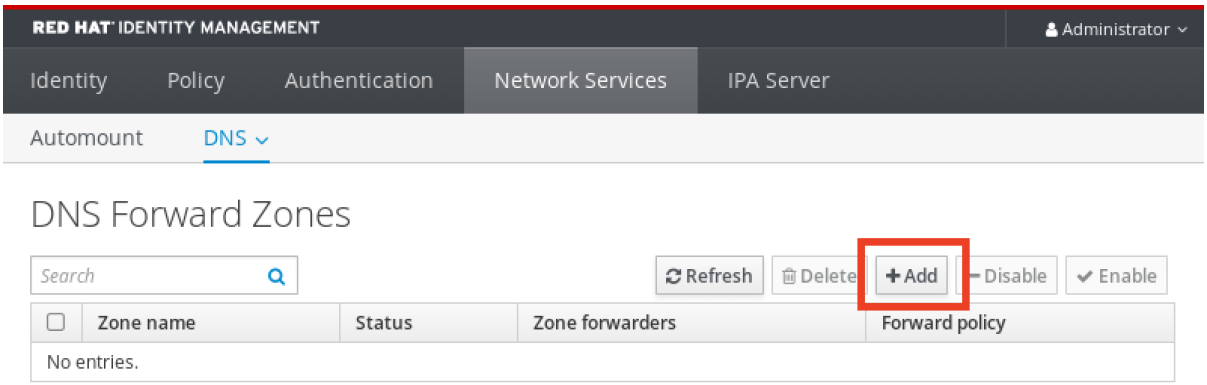
- 您知道要将查询转发到的 DNS 服务器的 Internet 协议(IP)地址。

**流程**

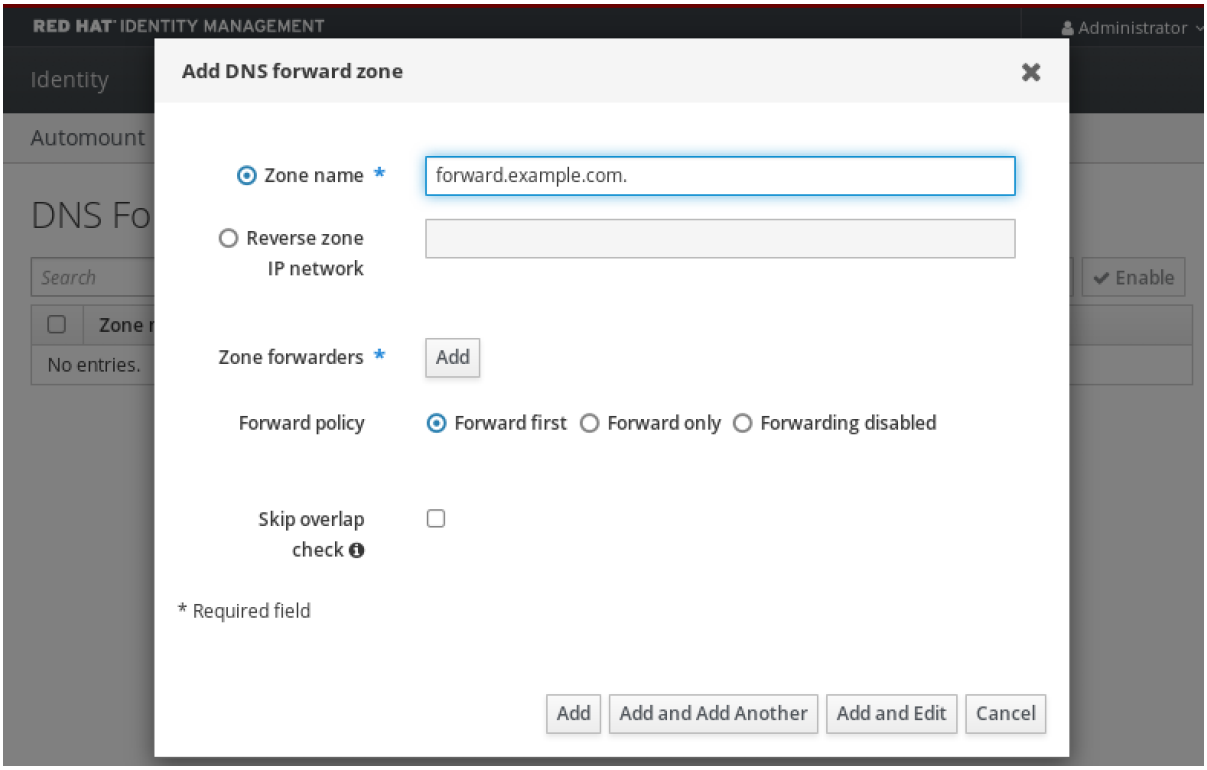
1. 在 IdM Web UI 中，选择 Network Services → DNS Forward Zones → DNS。



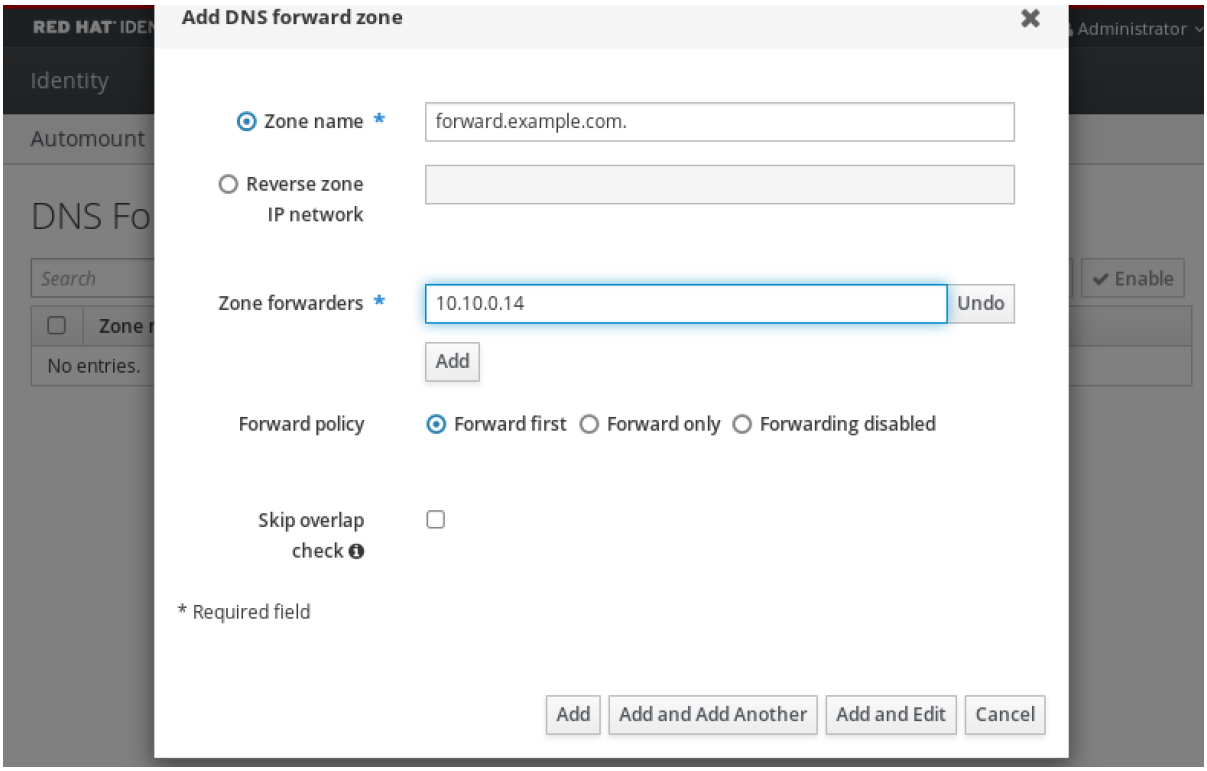

2. 在 DNS Forward Zones 部分，点 Add。



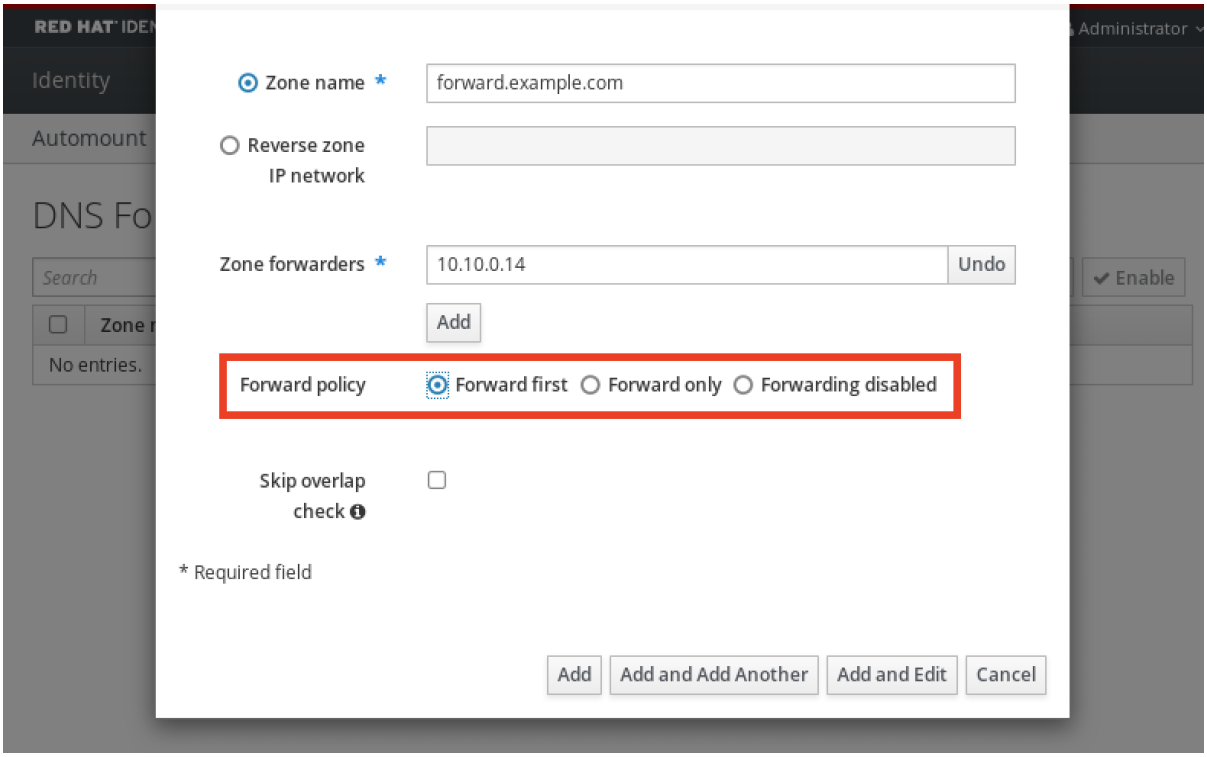

3. 在 Add DNS forward zone 窗口中，指定 forward zone 名称。




4. **点击 Add 按钮，并指定 DNS 服务器的 IP 地址来接收转发请求。您可以为每个转发区指定多个转发器。**



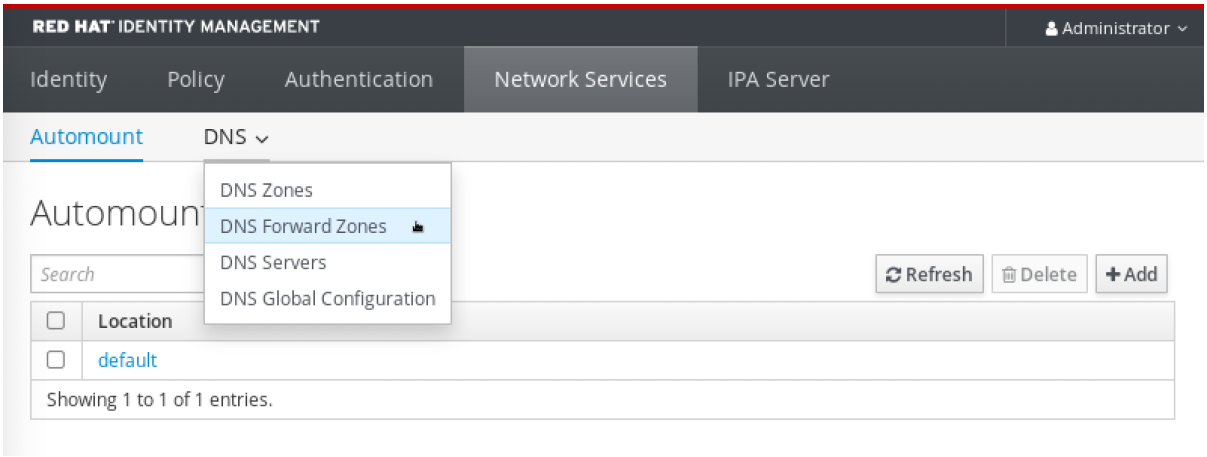

5. 选择 **Forward 策略。**



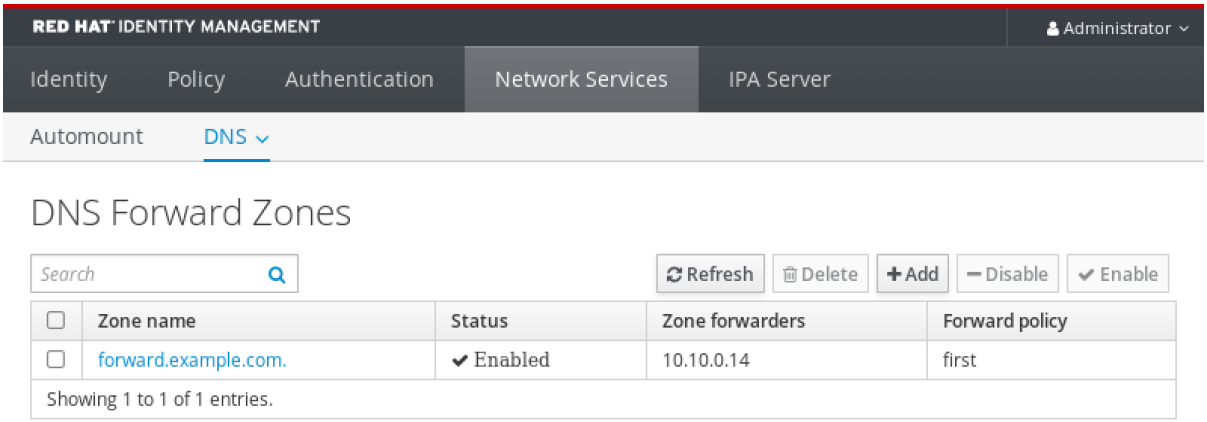

6. **单击窗口底部的 Add，以添加新的正向区域。**

**验证步骤**

1. **在 IdM Web UI 中，选择 Network Services → DNS Forward Zones → DNS。**




2. **验证您创建的 forward 区域（带有您指定的 forwarders 和 forward 策略）是否存在并在 IdM Web UI 中启用。**




## 96.6. 在 CLI 中添加 DNS 转发区域

按照以下流程使用命令行界面(CLI)添加 DNS 转发区。

**重要**

除非绝对需要，否则请不要使用转发区域。转发区域不是标准解决方案，使用它们可能会导致意外和有问题的行为。如果您必须使用 forward zone，限制使用它们覆盖全局转发配置。

在创建新 DNS 区域时，红帽建议您始终使用名称服务器(NS)记录和避免转发区域，始终使用标准 DNS 委派。在大多数情况下，使用全局转发器足够了，并且转发区不需要。

**先决条件**

- 以 IdM 管理员身份登录。
- 您知道要将查询转发到的 DNS 服务器的 Internet 协议(IP)地址。

**流程**

- 使用 dnsforwardzone-add 命令添加新的转发区域。如果 forward 策略 没有，请使用 --forwarder 选项指定至少一个 forwarder，并使用 --forward-policy 选项指定 forward 策略。

```
[user@server ~]$ ipa dnsforwardzone-add forward.example.com. --
forwarder=10.10.0.14 --forwarder=10.10.1.15 --forward-policy=first
```

```
Zone name: forward.example.com.
Zone forwarders: 10.10.0.14, 10.10.1.15
Forward policy: first
```

验证步骤

- 使用 dnsforwardzone-show 命令显示您刚才创建的 DNS 转发区域。

```
[user@server ~]$ ipa dnsforwardzone-show forward.example.com.

Zone name: forward.example.com.
Zone forwarders: 10.10.0.14, 10.10.1.15
Forward policy: first
```

## 96.7. 使用 ANSIBLE 在 IDM 中建立 DNS 全局转发器

按照以下流程，使用 Ansible playbook 在 IdM 中建立 DNS Global Forwarder。

在以下示例中，IdM 管理员创建 DNS 全局转发程序到端口 53，Internet 协议(IP)v4 地址为 8.8.6.6 和 IPv6 地址为 2001:4860:4860::8800 的 DNS 服务器。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 复制 set-configuration.yml Ansible playbook 文件。例如：

```
$ cp set-configuration.yml establish-global-forwarder.yml
```

4. 打开 create -global-forwarder.yml 文件进行编辑。

5. 通过设置以下变量来调整文件：

   a. 将 playbook 的 name 变量 更改为 Playbook，以在 IdM DNS 中建立全局转发器。

   b. 在 tasks 部分中，将任务 的名称 更改为 Create a DNS global forwarder 设为 8.8.6.6 和 2001:4860:4860::8800。

   c. 在 ipadnsconfig 部分的 forwarders 部分：

      i. 将第一个 ip_address 值更改为全局转发器的 IPv4 地址： 8.8.6.6。

      ii. 将第二个 ip_address 值更改为全局转发器的 IPv6 地址： 2001:4860:4860::8800。

iii. 验证 **端口** 值是否已设置为 **53**。

d. 将 **forward_policy** 更改为 **first**。

对于当前示例为修改过的 **Ansible playbook** 文件：

```
---
- name: Playbook to establish a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Create a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800
    ipadnsconfig:
      forwarders:
        - ip_address: 8.8.6.6
        - ip_address: 2001:4860:4860::8800
          port: 53
      forward_policy: first
      allow_sync_ptr: yes
```

6. 保存该文件。

7. 运行 **playbook**:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file establish-global-forwarder.yml
```

**其它资源**

- 请参阅 **/usr/share/doc/ansible-freeipa/** 目录中的 **README-dnsconfig.md** 文件。

## 96.8. 使用 ANSIBLE 确保 IDM 中存在 DNS 全局转发器

按照以下流程，使用 **Ansible playbook** 确保 **DNS** 全局转发器在 **IdM** 中存在。在下例中，**IdM** 管理员确保在 **DNS** 服务器中存在一个到 **IPv4** 地址为 **7.7.9.9**，**IP v6** 地址为 **2001:db8::1:0**，端口 **53** 的 **DNS global forwarder**。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 制作 forwarders-absent.yml Ansible playbook 文件的副本。例如：

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. 打开 ensure-presence-global-forwarder.yml 文件进行编辑。

5. 通过设置以下变量来调整文件：

   a. 将 playbook 的 name 变量 更改为 Playbook，以确保 IdM DNS 中存在全局转发器。

   b. 在 tasks 部分中，将任务 的名称 更改为确保 存在 DNS global forwarder 在端口 53 上存在 7.7.9.9 和 2001:db8::1:0。

   c. 在 ipadnsconfig 部分的 forwarders 部分：

      i. 将第一个 ip_address 值更改为全局转发器的 IPv4 地址： 7.7.9.9。

      ii. 将第二个 ip_address 值更改为全局转发器的 IPv6 地址： 2001:db8::1:0。

      iii. 验证 端口 值是否已设置为 53。

   d. 将 状态 更改为 present。

   对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Playbook to ensure the presence of a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0
on port 53
    ipadnsconfig:
      forwarders:
        - ip_address: 7.7.9.9
        - ip_address: 2001:db8::1:0
          port: 53
      state: present
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-of-a-global-forwarder.yml
```

**其它资源**

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsconfig.md 文件。

## 96.9. 使用 ANSIBLE 确保 IDM 中没有 DNS 全局转发器

按照以下流程，使用 Ansible playbook 确保 DNS 全局转发器在 IdM 中不存在。在以下示例流程中，IdM 管理员确保在端口 53 上没有互联网协议(IP)v4 地址为 8.8.6.6 和 IP v6 地址为 2001:4860:4860::8800 的 DNS 全局转发器。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 制作 forwarders-absent.yml Ansible playbook 文件的副本。例如：

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. 打开 ensure-absence-of-a-global-forwarder.yml 文件进行编辑。

5. 通过设置以下变量来调整文件：

   a. 将 playbook 的 name 变量 更改为 Playbook，以确保 IdM DNS 中缺少全局转发器。

   b. 在 tasks 部分中，将任务 的名称 更改为确保 没有 DNS 全局转发器在端口 53 上为 8.8.6.6 和 2001:4860:4860::8800。

   c. 在 ipadnsconfig 部分的 forwarders 部分：

      i. 将第一个 ip_address 值更改为全局转发器的 IPv4 地址： 8.8.6.6。

      ii. 将第二个 ip_address 值更改为全局转发器的 IPv6 地址： 2001:4860:4860::8800。

      iii. 验证 端口 值是否已设置为 53。

a.
将 action 变量设置为 member。

e.
验证 state 已设为 absent。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Playbook to ensure the absence of a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the absence of a DNS global forwarder to 8.8.6.6 and
2001:4860:4860::8800 on port 53
    ipadnsconfig:
      forwarders:
        - ip_address: 8.8.6.6
        - ip_address: 2001:4860:4860::8800
          port: 53
      action: member
      state: absent
```

重要

如果您仅在 playbook 中使用 state: absent 选项，而不使用 action: member，则 playbook 会失败。

6.
保存该文件。

7.
运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-
absence-of-a-global-forwarder.yml
```

其它资源

- /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsconfig.md 文件

- ipadnsconfig ansible-freeipa 模块中的 action: member 选项

## 96.10. 使用 ANSIBLE 确保 DNS 全局转发器在 IDM 中被禁用

按照以下流程，使用 Ansible playbook 确保 DNS Global Forwarders 在 IdM 中被禁用。在以下示例中，IdM 管理员确保将全局转发器的转发策略设置为 none，这样可有效地禁用全局转发器。

### 先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

### 流程

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

-

```
[ipaserver]
server.idm.example.com
```

3. 验证 disable-global-forwarders.yml Ansible playbook 文件的内容，它已配置为禁用所有 DNS 全局转发器。例如：

```
$ cat disable-global-forwarders.yml
---
- name: Playbook to disable global DNS forwarders
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Disable global forwarders.
    ipadnsconfig:
      forward_policy: none
```

4. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disable-global-forwarders.yml
```

**其它资源**

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsconfig.md 文件。

### 96.11. 使用 ANSIBLE 确保 IDM 中存在 DNS 转发区域

按照以下流程，使用 Ansible playbook 确保 DNS 转发区域在 IdM 中存在。在以下示例中，IdM 管理员确保 example.com 的 DNS 转发区域存在到 Internet 协议(IP)地址为 8.8.8.8 的 DNS 服务器。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。

- 示例假定在 *~/MyPlaybooks/* 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 制作 forwarders-absent.yml Ansible playbook 文件的副本。例如：

```
$ cp forwarders-absent.yml ensure-presence-forwardzone.yml
```

4. 打开 ensure-presence-forwardzone.yml 文件进行编辑。

5. 通过设置以下变量来调整文件：

   a. 将 playbook 的 name 变量 更改为 Playbook，以确保 IdM DNS 中存在 dnsforwardzone。

   b. 在 tasks 部分中，将任务 的名称 更改为 Ensure of a example.com 的

dnsforwardzone to 8.8.8.8。

c. 在 tasks 部分中，将 ipadnsconfig 标题改为 ipadnsforwardzone。

d. 在 ipadnsforwardzone 部分：

i. 添加 ipaadmin_password 变量，并将其设置为 IdM 管理员密码。

ii. 添加 name 变量，并将它设置为 example.com。

iii. 在 forwarders 部分中：

A. 删除 ip_address 和 port 行。

B. 通过在短划线后指定 DNS 服务器的 IP 地址来添加 DNS 服务器的 IP 地址以接收转发的请求：

```
- 8.8.8.8
```

iv. 添加 forwardpolicy 变量，并将它设为 第一。

v. 添加 skip_overlap_check 变量，并将它设为 true。

vi. 将 state 变量更改为 present。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Playbook to ensure the presence of a dnsforwardzone in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
```

```
tasks:
- name: Ensure the presence of a dnsforwardzone for example.com to 8.8.8.8
ipadnsforwardzone:
   ipaadmin_password: "{{ ipaadmin_password }}"
   name: example.com
   forwarders:
      - 8.8.8.8
   forwardpolicy: first
   skip_overlap_check: true
   state: present
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-forwardzone.yml
```

**其它资源**

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsforwardzone.md 文件。

### 96.12. 使用 ANSIBLE 确保 DNS 转发区域 在 IDM 中有多个转发器

按照以下流程，使用 Ansible playbook 确保 IdM 中的 DNS Forward Zone 有多个转发器。在以下示例中，IdM 管理员确保 example .com 的 DNS 转发区转发到 8.8.8.8 和 4.4.4.4。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 制作 forwarders-absent.yml Ansible playbook 文件的副本。例如：

```
$ cp forwarders-absent.yml ensure-presence-multiple-forwarders.yml
```

4. 打开 ensure-presence-multiple-forwarders.yml 文件进行编辑。

5. 通过设置以下变量来调整文件：

   a. 将 playbook 的 name 变量 更改为 Playbook，以确保 IdM DNS 中的 dnsforwardzone 中存在多个转发器。

   b. 在 tasks 部分中，将任务 的名称 更改为在 example .com 的 dnsforwardzone 中确保存在 8.8.8.8 和 4.4.4.4 forwarders。

c. 在 tasks 部分中，将 ipadnsconfig 标题改为 ipadnsforwardzone。

d. 在 ipadnsforwardzone 部分：

i. 添加 ipaadmin_password 变量，并将其设置为 IdM 管理员密码。

ii. 添加 name 变量，并将它设置为 example.com。

iii. 在 forwarders 部分中：

A. 删除 ip_address 和 port 行。

B. 添加您要确保的 DNS 服务器的 IP 地址，以短划线开头：

```
- 8.8.8.8
- 4.4.4.4
```

iv. 将 state 变量更改为 present。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: name: Playbook to ensure the presence of multiple forwarders in a
dnsforwardzone in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure presence of 8.8.8.8 and 4.4.4.4 forwarders in dnsforwardzone for
example.com
  ipadnsforwardzone:
     ipaadmin_password: "{{ ipaadmin_password }}"
    name: example.com
     forwarders:
        - 8.8.8.8
        - 4.4.4.4
     state: present
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-multiple-forwarders.yml
```

其它资源

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsforwardzone.md 文件。

## 96.13. 使用 ANSIBLE 确保 IDM 中 DNS FORWARD 区域被禁用

按照以下流程，使用 Ansible playbook 确保 DNS Forward Zone 在 IdM 中被禁用。在以下示例中，IdM 管理员确保 example.com 的 DNS 转发区被禁用。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 制作 forwarders-absent.yml Ansible playbook 文件的副本。例如：

```
$ cp forwarders-absent.yml ensure-disabled-forwardzone.yml
```

4. 打开 ensure-disabled-forwardzone.yml 文件进行编辑。

5. 通过设置以下变量来调整文件：

   a. 将 playbook 的 name 变量 更改为 Playbook，以确保在 IdM DNS 中禁用了 dnsforwardzone。

   b. 在 tasks 部分中，将任务 的名称 更改为 Ensure a dnsforwardzone for example.com。

   c. 在 tasks 部分中，将 ipadnsconfig 标题改为 ipadnsforwardzone。

   d. 在 ipadnsforwardzone 部分：

      i. 添加 ipaadmin_password 变量，并将其设置为 IdM 管理员密码。

      ii. 添加 name 变量，并将它设置为 example.com。

iii. 删除整个 forwarders 部分。

iv. 将 state 变量更改为 disabled。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Playbook to ensure a dnsforwardzone is disabled in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure a dnsforwardzone for example.com is disabled
  ipadnsforwardzone:
     ipaadmin_password: "{{ ipaadmin_password }}"
     name: example.com
     state: disabled
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-disabled-forwardzone.yml
```

**其它资源**

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsforwardzone.md 文件。

### 96.14. 使用 ANSIBLE 确保 IDM 中没有 DNS 转发区域

按照以下流程，使用 Ansible playbook 确保 DNS 转发区域在 IdM 中不存在。在以下示例中，IdM 管理员确保 example.com 缺少 DNS 转发区。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：

- 您使用 Ansible 版本 2.14 或更高版本。
- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 制作 forwarders-absent.yml Ansible playbook 文件的副本。例如：

```
$ cp forwarders-absent.yml ensure-absence-forwardzone.yml
```

4. 打开 ensure-absence-forwardzone.yml 文件进行编辑。

5. 通过设置以下变量来调整文件：

a. 将 playbook 的 name 变量 更改为 Playbook，以确保 IdM DNS 中没有 dnsforwardzone。

b. 在 tasks 部分中，将任务 的名称 更改为 Ensure the dnsforwardzone for example.com。

c. 在 tasks 部分中，将 ipadnsconfig 标题改为 ipadnsforwardzone。

d. 在 ipadnsforwardzone 部分：

i. 添加 ipaadmin_password 变量，并将其设置为 IdM 管理员密码。

ii. 添加 name 变量，并将它设置为 example.com。

iii. 删除整个 forwarders 部分。

iv. 将 state 变量保留为 absent。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Playbook to ensure the absence of a dnsforwardzone in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the absence of a dnsforwardzone for example.com
    ipadnsforwardzone:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: example.com
      state: absent
```

6. 保存该文件。

7. 运行 **playbook:**

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-forwardzone.yml
```

**其它资源**

- 请参阅 **/usr/share/doc/ansible-freeipa/** 目录中的 **README-dnsforwardzone.md** 文件。

# 第 97 章 在 IDM 中管理 DNS 记录

本章论述了如何在身份管理(IdM)中管理 DNS 记录。作为 IdM 管理员，您可以在 IdM 中添加、修改和删除 DNS 记录。本章包含以下部分：

- IdM 中的 DNS 记录
- 从 IdM Web UI 添加 DNS 资源记录
- 通过 IdM CLI 添加 DNS 资源记录
- 常用 ipa dnsrecord-add 选项
- 删除 IdM Web UI 中的 DNS 记录
- 在 IdM Web UI 中删除整个 DNS 记录
- 删除 IdM CLI 中的 DNS 记录

**先决条件**

- 您的 IdM 部署包含一个集成的 DNS 服务器。有关如何使用集成 DNS 安装 IdM 的详情，请查看以下链接之一：
  - 安装 IdM 服务器：使用集成的 DNS，使用集成的 CA 作为 root CA。
  - 安装 IdM 服务器：使用集成的 DNS，使用外部 CA 作为 root CA。

## 97.1. IDM 中的 DNS 记录

身份管理(IdM)支持许多不同的 DNS 记录类型。以下四个最常使用：

**A**

这是主机名和 IPv4 地址的基本映射。A 记录的记录名称是主机名，如 www。A 记录的 IP Address 值是一个 IPv4 地址，如 192.0.2.1。

有关 A 记录的更多信息，请参阅 RFC 1035。

**AAAA**

这是主机名和 IPv6 地址的基本映射。AAAA 记录的记录名称是主机名，如 www。IP Address 值是一个 IPv6 地址，如 2001:DB8::1111。

有关 AAAA 记录的更多信息，请参阅 RFC 3596。

**SRV**

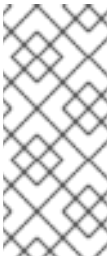
*服务(SRV)资源记录将服务名称映射到* 提供该特定服务的服务器的 DNS 名称。例如，此记录类型可将 LDAP 目录等服务映射到管理此目录的服务器。

SRV 记录的记录名称格式为 *_service._protocol*，如 _ldap._tcp。SRV 记录的配置选项包括目标服务的优先级、权重、端口号和主机名。

有关 SRV 记录的详情请参考 RFC 2782。

**PTR**

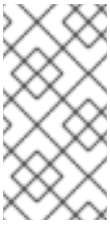
指针记录(PTR)添加反向 DNS 记录，它将 IP 地址映射到域名。



**注意**

IPv4 地址的所有反向 DNS 查找都使用在 in-addr.arpa. 域中定义的反向条目。反向地址（人类可读形式）与常规 IP 地址正好相反，其中 in-addr.arpa. 域附加到该地址。例如，对于网络地址 192.0.2.0/24，反向区域为 2.0.192.in-addr.arpa。

PTR 的记录名称必须采用 RFC 1035 中指定的标准格式，以 RFC 23 17 和 RFC 3596 扩展。主机名值必须是您要为其创建记录的主机的规范主机名。



**注意**

也可以为 IPv6 地址配置反向区域，包括 .ip6.arpa. 域中的区域。有关 IPv6 反向区的更多信息，请参阅 RFC 3596。

添加 DNS 资源记录时，请注意很多记录需要不同的数据。例如，CNAME 记录需要主机名，而 A 记录则需要 IP 地址。在 IdM Web UI 中，用于添加新记录的表单中的字段会自动更新，以反映当前所选记录类型所需的数据。

### 97.2. 在 IDM WEB UI 中添加 DNS 资源记录

按照以下流程在身份管理(IdM) Web UI 中添加 DNS 资源记录。

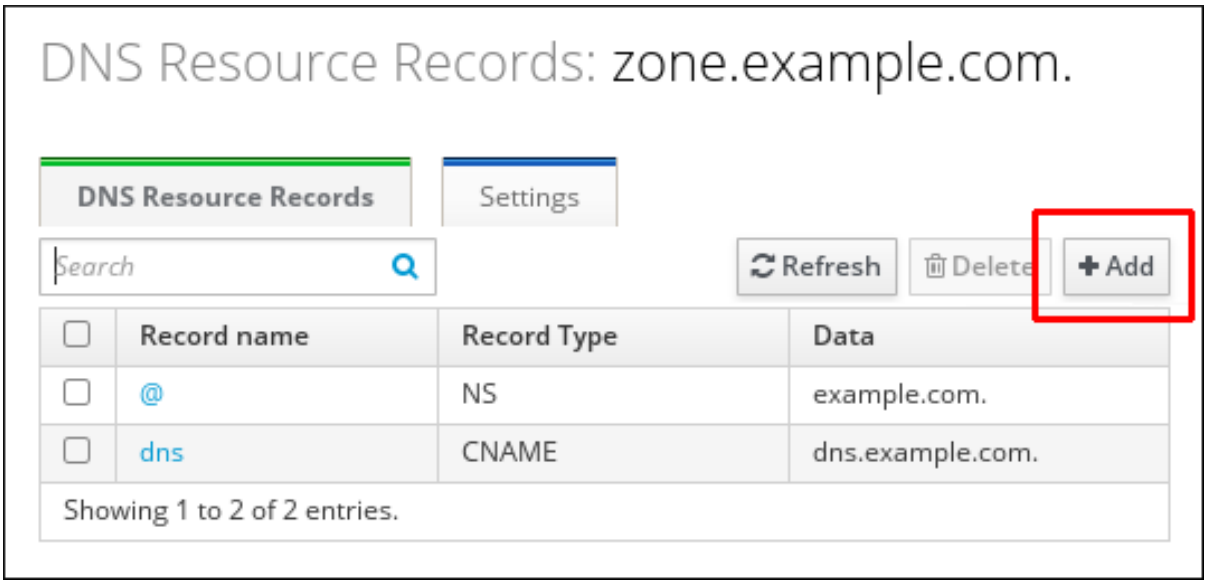
**先决条件**

- 要添加 DNS 记录的 DNS 区域存在，并由 IdM 管理。有关在 IdM DNS 中创建 DNS 区域的更多信息，请参阅在 IdM 中管理 DNS 区。
- 以 IdM 管理员身份登录。

**流程**

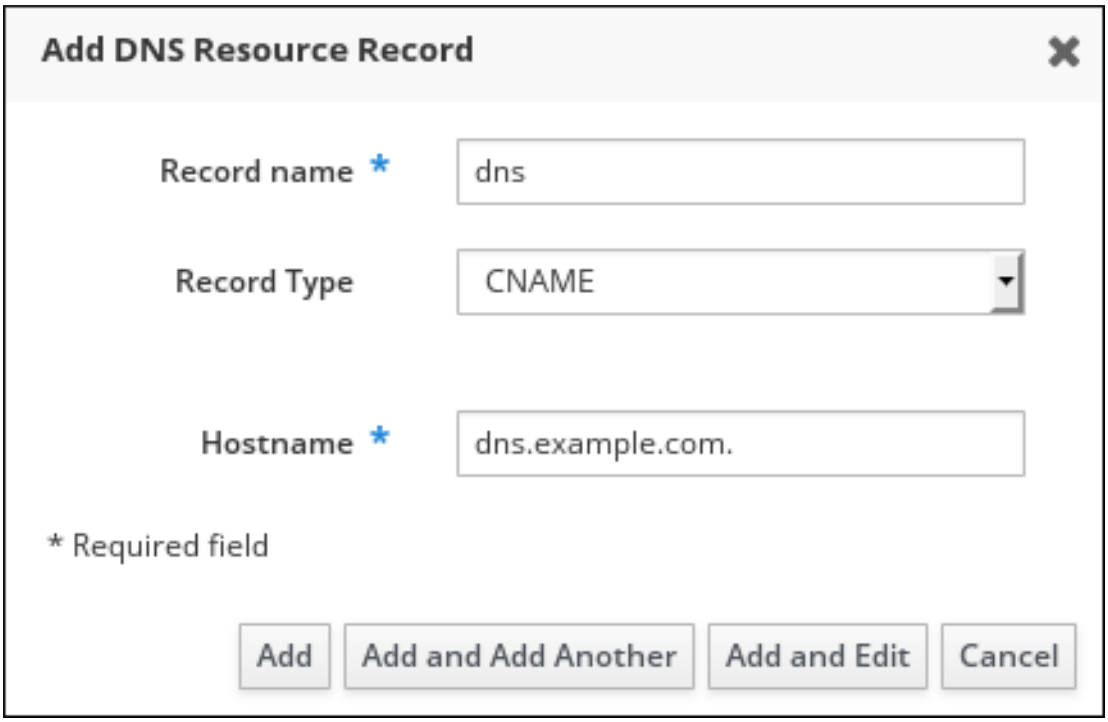
1. 在 IdM Web UI 中，点击 Network Services → DNS → DNS Zones。
2. 点击您要添加 DNS 记录的 DNS 区域。
3. 在 DNS Resource Records 部分，点 Add 来添加新记录。

图 97.1. 添加新 DNS 资源记录




4. 选择要创建的记录类型，并根据需要填写其他字段。

图 97.2. 定义新的 DNS 资源记录




5. 单击 **Add** 以确认新记录。

## 97.3. 通过 IDM CLI 添加 DNS 资源记录

**按照以下流程，通过命令行界面(CLI)添加任何类型的 DNS 资源记录。**

**先决条件**

- 您要添加 DNS 记录的 DNS 区域存在。有关在 IdM DNS 中创建 DNS 区域的更多信息，请参阅在 IdM 中管理 DNS 区。
- 以 IdM 管理员身份登录。

**流程**

1. 要添加 DNS 资源记录，请使用 ipa dnsrecord-add 命令。该命令采用以下语法：

```
$ ipa dnsrecord-add zone_name record_name --record_type_option=data
```

在以上命令中：

- *zone_name* 是正在向其添加记录的 DNS 区域的名称。
- *record_name* 是新 DNS 资源记录的标识符。

例如，要将 host1 的 A 类型 DNS 记录添加到 idm.example.com 区域，请输入：

```
$ ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123
```

## 97.4. COMMON IPA DNSRECORD-* 选项

在身份管理(IdM)中添加、修改和删除最常见的 DNS 资源记录类型时，您可以使用以下选项：

- A (IPv4)
- AAAA (IPv6)
- SRV

- **PTR**

**在 Bash 中，您可以通过在大括号内列出逗号分隔列表中的值来定义多个条目，如 --option={val1,val2,val3}。**

**表 97.1. 常规记录选项**

| *选项* | *描述* |
|---|---|
| **--ttl**=*number* | 为记录设置生存时间。 |
| **--structured** | 解析原始 DNS 记录，并以结构化格式返回它们。 |

**表 97.2. "a" 记录选项**

| *选项* | *描述* | *示例* |
|---|---|---|
| **--a-rec**=*ARECORD* | 传递单个 A 记录或 A 记录列表. | **ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123** |
| | 可以使用给定 IP 地址创建通配符 A 记录. | **ipa dnsrecord-add idm.example.com "*" --a-rec=192.168.122.123** [a] |
| **--a-ip-address**=*string* | 提供 记录的 IP 地址。在创建记录时，指定 **A** 记录值的选项为 **--a-rec**。但是，修改 **A** 记录时，**--a-rec** 选项用于指定 **A** 记录的当前值。使用 **--a-ip-address** 选项设置新值。 | **ipa dnsrecord-mod idm.example.com --a-rec 192.168.122.123 --a-ip-address 192.168.122.124** |

[a] 这个示例创建了一个通配符 **A** 记录，其 IP 地址为 192.0.2.123。

**表 97.3. "AAAA"记录选项**

| *选项* | *描述* | *示例* |
|---|---|---|
| **--aaaa-rec**=*AAAARECORD* | 传递单个 AAAA(IPv6)记录或 AAAA 记录列表。 | **ipa dnsrecord-add idm.example.com www --aaaa-rec 2001:db8::1231:5675** |
| **--aaaa-ip-address**=*string* | 提供 记录的 IPv6 地址。在创建记录时，指定 **A** 记录值的选项为 **--aaaa-rec**。但是，修改 **A** 记录时，**--aaaa-rec** 选项用于指定 **A** 记录的当前值。使用 **--a-ip-address** 选项设置新值。 | **ipa dnsrecord-mod idm.example.com --aaaa-rec 2001:db8::1231:5675 --aaaa-ip-address 2001:db8::1231:5676** |

| 选项 | 描述 | 示例 |
| --- | --- | --- |

**表 97.4. "PTR"记录选项**

| 选项 | 描述 | 示例 |
| --- | --- | --- |
| **--ptr-rec**=*PTRRECORD* | 传递单个 PTR 记录或 PTR 记录列表。添加反向 DNS 记录时，与添加其他 DNS 记录的用法不同，与 **ipa dnsrecord-add** 命令一起使用的区域名称相反。通常，主机 IP 地址是给定网络中 IP 地址的最后一个八进制数。右侧的第一个示例为 server4.idm.example.com 添加 PTR 记录，其 IPv4 地址 为 192.168.122.4。第二个示例在 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa **中添加一个反向 DNS 条目。**主机 server2.example.com 的 IPv6 反向区域，IP 地址为 2001:DB8::1111. | **ipa dnsrecord-add 122.168.192.in-addr.arpa 4 --ptr-rec server4.idm.example.com.**<br>**$ ipa dnsrecord-add 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.1.1.1.0.0.0.0.0.0.0.0.0.0.0 --ptr-rec server2.idm.example.com.** |
| **--ptr-hostname**=*string* | 提供记录的主机名。 | |

**表 97.5. "SRV"记录选项**

| 选项 | 描述 | 示例 |
| --- | --- | --- |
| **--srv-rec**=*SRVRECORD* | 传递单个 SRV 记录或 SRV 记录列表。在右侧的示例中，_ldap._tcp 定义 SRV 记录的服务类型和连接协议。**srv-rec** 选项定义优先级、权重、端口和目标值。示例中的权重值为 51 和 49（总和为 100），它们代表使用特定记录的可能性（以百分比表示）。 | **# ipa dnsrecord-add idm.example.com _ldap._tcp --srv-rec="0 51 389 server1.idm.example.com."**<br>**# IPA dnsrecord-add server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com."** |
| **--srv-priority**=*number* | 设置记录的优先级。某一服务类型可以有多个 SRV 记录。优先级(0 - 65535)设置记录的排名；数值越低，优先级越高。服务必须首先使用优先级最高的记录。 | **# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com." --srv-priority=0** |

| *选项* | *描述* | *示例* |
| --- | --- | --- |
| **--srv-weight**=*number* | 设置记录的权重。这有助于确定优先级相同的 SRV 记录的顺序。集合权重应加到 100，代表使用特定记录的可能性（百分比）。 | **# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 49 389 server2.idm.example.com." --srv-weight=60** |
| **--srv-port**=*number* | 指定目标主机上 服务的端口。 | **# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 60 389 server2.idm.example.com." --srv-port=636** |
| **--srv-target**=*string* | 指定目标主机的域名。如果域中的服务不可用，这可以是单个句点(.)。 | |

**其它资源**

- 运行 **ipa dnsrecord-add --help**。

### 97.5. 删除 IDM WEB UI 中的 DNS 记录

**按照以下流程，使用 IdM Web UI 删除身份管理(IdM)中的 DNS 记录。**

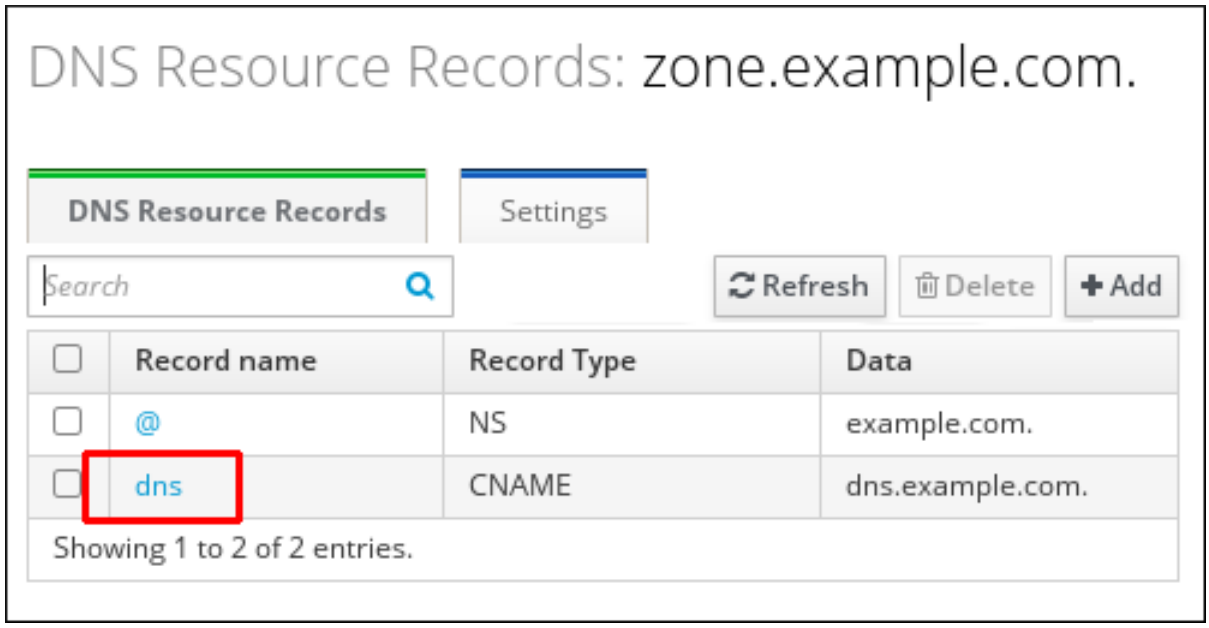
**先决条件**

- **以 IdM 管理员身份登录。**

**流程**

1. **在 IdM Web UI 中，点击 Network Services → DNS → DNS Zones。**
2. **单击您要从中删除 DNS 记录的区域，如 example.com。**
3. **在 DNS Resource Records 部分，点击资源记录的名称。**

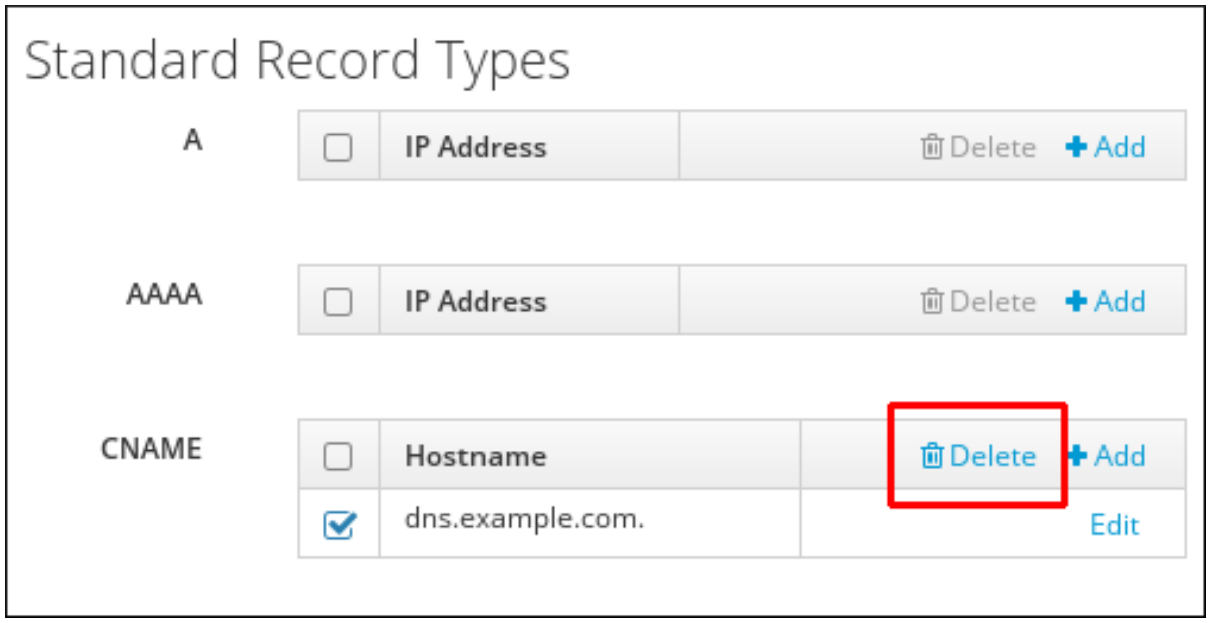
图 97.3. 选择 DNS 资源记录




4. 按要删除的记录类型的名称选择复选框。

5. 点 Delete。

图 97.4. 删除 DNS 资源记录




现在，所选的记录类型已被删除。资源记录的其他配置保持不变。

其它资源

- 请参阅 在 IdM Web UI 中删除整个 DNS 记录。

## 97.6. 在 IDM WEB UI 中删除整个 DNS 记录

按照以下流程，使用身份管理(IdM) Web UI 删除区域中特定资源的所有记录。

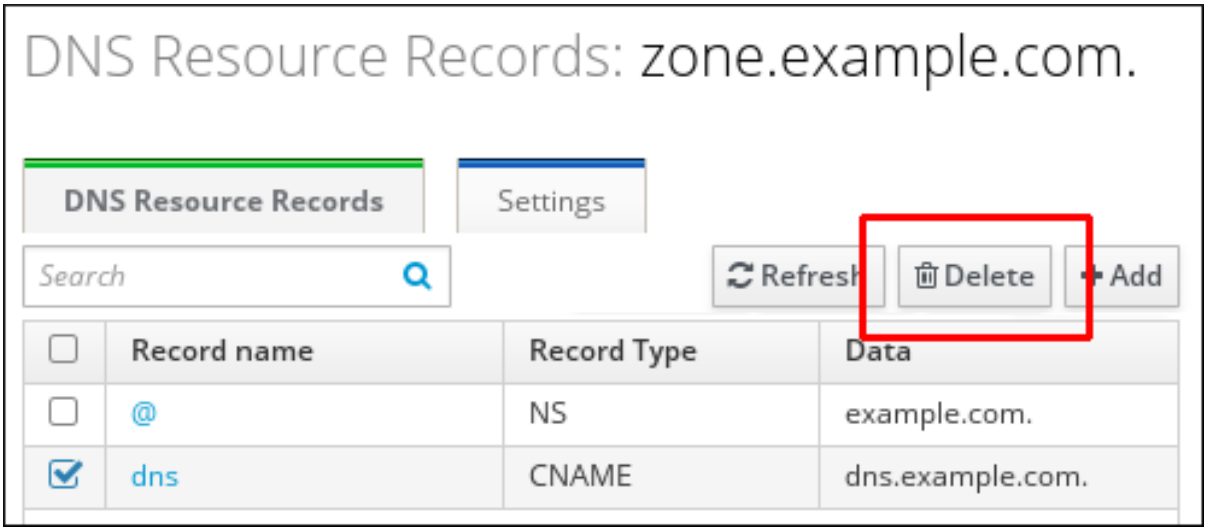
### 先决条件

- 以 IdM 管理员身份登录。

### 流程

1. 在 IdM Web UI 中，点击 Network Services → DNS → DNS Zones。
2. 单击您要从中删除 DNS 记录的区域，如 zone.example.com。
3. 在 DNS Resource Records 部分，选中要删除的资源记录的复选框。
4. 点删除。

图 97.5. 删除 Entire 资源记录




现在，整个资源记录已被删除。

## 97.7. 删除 IDM CLI 中的 DNS 记录

按照以下流程，从身份管理(IdM) DNS 管理的区中删除 DNS 记录。

**先决条件**

- 以 IdM 管理员身份登录。

**流程**

- 要从区中删除记录，请使用 ipa dnsrecord-del 命令，并将 *--recordType*-rec 选项与记录值一起添加。例如，要删除 A 类型记录：

```
$ ipa dnsrecord-del example.com www --a-rec 192.0.2.1
```

如果您在没有任何选项的情况下运行 ipa dnsrecord-del，该命令会提示输入要删除的记录的信息。请注意，通过 命令传递 --del-all 选项将删除区域的所有相关记录。

**其它资源**

- 运行 ipa dnsrecord-del --help 命令。

## 97.8. 其它资源

- 请参阅 在 IdM 中 使用 Ansible 管理 DNS 记录。

# 第 98 章 在使用外部 DNS 时，以系统方式更新 DNS 记录

使用外部 DNS 时，身份管理(IdM)不会在拓扑更改后自动更新 DNS 记录。您可以以系统方式更新由外部 DNS 服务管理的 DNS 记录，从而减少了手动 DNS 更新的需求。

更新 DNS 记录会删除旧的或无效的 DNS 记录，并添加新记录。您必须在拓扑更改后更新 DNS 记录，例如：

- 安装或卸载副本后
- 在 IdM 服务器上安装 CA、DNS、KRA 或 Active Directory 信任后

## 98.1. 使用 GUI 更新外部 DNS 记录

如果您对拓扑进行任何更改，则必须使用外部 DNS GUI 更新外部 DNS 记录。

**流程**

1. 显示您必须更新的记录：

```
$ ipa dns-update-system-records --dry-run
 IPA DNS records:
   _kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88 ipa.example.com.
   _kerberos-master._udp.example.com. 86400 IN SRV 0 100 88 ipa.example.com.
[... output truncated ...]
```

2. 使用外部 DNS GUI 更新记录。

## 98.2. 使用 NSUPDATE 更新外部 DNS 记录

您可以使用 nsupdate 工具更新外部 DNS 记录。您还可以将命令添加到脚本中以自动化进程。要使用 nsupdate 工具更新，您需要使用 DNS 记录生成文件，然后继续发送使用 TSIG 保护的 nsupdate 请求，或者发送使用 GSS-TSIG 保护的 nsupdate 请求。

**流程**

-

要为 nsupdate 生成 DNS 记录的文件，请使用 'ipa dns-update-system-records --dry-run 命令及 --out 选项。--out 选项指定要生成的文件的路径：

```
$ ipa dns-update-system-records --dry-run --out dns_records_file.nsupdate
 IPA DNS records:
   _kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88 ipa.example.com.
   _kerberos-master._udp.example.com. 86400 IN SRV 0 100 88 ipa.example.com.
[... output truncated ...]
```

生成的文件包含 nsupdate 工具接受的格式所需的 DNS 记录。

- 生成的记录取决于：
  - 自动检测记录要更新的区域。
  - 自动检测区域的权威服务器。

    如果您使用一个样式的 DNS 设置，或者缺少区委派，则 nsupdate 可能无法找到正确的区域和服务器。在这种情况下，在生成的文件的开头添加以下选项：

  - server ：指定 nsupdate 将记录的权威 DNS 服务器的服务器名称或端口。
  - zone ：指定 nsupdate 放置记录的区域名称。

**例 98.1. 生成的记录**

```
$ cat dns_records_file.nsupdate
zone example.com.
server 192.0.2.1
; IPA DNS records
update delete _kerberos-master._tcp.example.com. SRV
update add _kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88
ipa.example.com.
[... output truncated ...]
```

## 98.3. 发送使用 TSIG 保护的 NSUPDATE 请求

使用 nsupdate 发送请求时，请确保正确保护它。事务签名(TSIG)可让您将 nsupdate 与共享密钥一起使用。

**先决条件**

- 您必须为 TSIG 配置您的 DNS 服务器。
- DNS 服务器及其客户端必须具有共享密钥。

**流程**

- 运行 nsupdate 命令并使用以下选项之一提供共享 secret：
  - -k 提供 TSIG 身份验证密钥：

```
$ nsupdate -k tsig_key.file dns_records_file.nsupdate
```

  - -y 从密钥名称和以 Base64 编码的共享 secret 生成签名：

```
$ nsupdate -y algorithm:keyname:secret dns_records_file.nsupdate
```

## 98.4. 发送使用 GSS-TSIG 保护的 NSUPDATE 请求

使用 nsupdate 发送请求时，请确保正确保护它。GSS-TSIG 使用 GSS-API 接口来获取 secret TSIG 密钥。GSS-TSIG 是 TSIG 协议的扩展。

**先决条件**

- 必须为您的 GSS-TSIG 配置您的 DNS 服务器。

**注意**

此流程假设 Kerberos V5 协议被用作 GSS-API 的技术。

**流程**

1. 使用允许更新记录的主体进行身份验证：

```
$ kinit principal_allowed_to_update_records@REALM
```

2. 使用 -g 选项运行 nsupdate 以启用 GSS-TSIG 模式：

```
$ nsupdate -g dns_records_file.nsupdate
```

### 98.5. 其它资源

- nsupdate (8) 手册页
- RFC 2845 描述了 TSIG 协议
- RFC 3645 描述了 GSS-TSIG 算法

# 第 99 章 使用 ANSIBLE 管理 IDM 中的 DNS 记录

本章论述了如何使用 Ansible playbook 管理身份管理(IdM)中的 DNS 记录。作为 IdM 管理员，您可以在 IdM 中添加、修改和删除 DNS 记录。本章包含以下部分：

- 确保使用 Ansible 在 IdM 中存在 A 和 AAAA DNS 记录
- 确保使用 Ansible 在 IdM 中存在 A 和 PTR DNS 记录
- 确保使用 Ansible 在 IdM 中存在多个 DNS 记录
- 确保使用 Ansible 在 IdM 中存在多个 CNAME 记录
- 使用 Ansible 在 IdM 中存在 SRV 记录

## 99.1. IDM 中的 DNS 记录

身份管理(IdM)支持许多不同的 DNS 记录类型。以下四个最常使用：

**A**

这是主机名和 IPv4 地址的基本映射。A 记录的记录名称是主机名，如 www。A 记录的 IP Address 值是一个 IPv4 地址，如 192.0.2.1。

有关 A 记录的更多信息，请参阅 RFC 1035。

**AAAA**

这是主机名和 IPv6 地址的基本映射。AAAA 记录的记录名称是主机名，如 www。IP Address 值是一个 IPv6 地址，如 2001:DB8::1111。

有关 AAAA 记录的更多信息，请参阅 RFC 3596。

**SRV**

*服务(SRV)资源记录将服务名称映射到* 提供该特定服务的服务器的 DNS 名称。例如，此记录类型可将 LDAP 目录等服务映射到管理此目录的服务器。

SRV 记录的记录名称格式为 *_service._protocol*，如 _ldap._tcp。SRV 记录的配置选项包括目标服务的优先级、权重、端口号和主机名。

有关 SRV 记录的详情请参考 RFC 2782。

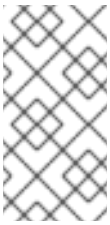
**PTR**

指针记录(PTR)添加反向 DNS 记录，它将 IP 地址映射到域名。

**注意**

IPv4 地址的所有反向 DNS 查找都使用在 in-addr.arpa. 域中定义的反向条目。反向地址（人类可读形式）与常规 IP 地址正好相反，其中 in-addr.arpa. 域附加到该地址。例如，对于网络地址 192.0.2.0/24，反向区域为 2.0.192.in-addr.arpa。

PTR 的记录名称必须采用 RFC 1035 中指定的标准格式，以 RFC 23 17 和 RFC 3596 扩展。主机名值必须是您要为其创建记录的主机的规范主机名。



**注意**

也可以为 IPv6 地址配置反向区域，包括 .ip6.arpa. 域中的区域。有关 IPv6 反向区的更多信息，请参阅 RFC 3596。

添加 DNS 资源记录时，请注意很多记录需要不同的数据。例如，CNAME 记录需要主机名，而 A 记录则需要 IP 地址。在 IdM Web UI 中，用于添加新记录的表单中的字段会自动更新，以反映当前所选记录类型所需的数据。

### 99.2. COMMON IPA DNSRECORD-* 选项

在身份管理(IdM)中添加、修改和删除最常见的 DNS 资源记录类型时，您可以使用以下选项：

- A (IPv4)

- **AAAA (IPv6)**
- **SRV**
- **PTR**

**在 Bash 中，您可以通过在大括号内列出逗号分隔列表中的值来定义多个条目，如 --option={val1,val2,val3}。**

**表 99.1. 常规记录选项**

| *选项* | *描述* |
|---|---|
| **--ttl**=*number* | 为记录设置生存时间。 |
| **--structured** | 解析原始 DNS 记录，并以结构化格式返回它们。 |

**表 99.2. "a" 记录选项**

| *选项* | *描述* | *示例* |
|---|---|---|
| **--a-rec**=*ARECORD* | 传递单个 A 记录或 A 记录列表. | **ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123** |
| | 可以使用给定 IP 地址创建通配符 A 记录. | **ipa dnsrecord-add idm.example.com "*" --a-rec=192.168.122.123** [a] |
| **--a-ip-address**=*string* | 提供 记录的 IP 地址。在创建记录时，指定 **A** 记录值的选项为 **--a-rec**。但是，修改 **A** 记录时，**--a-rec** 选项用于指定 **A** 记录的当前值。使用 **--a-ip-address** 选项设置新值。 | **ipa dnsrecord-mod idm.example.com --a-rec 192.168.122.123 --a-ip-address 192.168.122.124** |

[a] 这个示例创建了一个通配符 **A** 记录，其 IP 地址为 192.0.2.123。

**表 99.3. "AAAA"记录选项**

| 选项 | 描述 | 示例 |
| --- | --- | --- |
| **--aaaa-rec**=*AAAARECORD* | 传递单个 AAAA(IPv6)记录或 AAAA 记录列表。 | **ipa dnsrecord-add idm.example.com www --aaaa-rec 2001:db8::1231:5675** |
| **--aaaa-ip-address**=*string* | 提供 记录的 IPv6 地址。在创建记录时，指定 **A** 记录值的选项为 **--aaaa-rec**。但是，修改 **A** 记录时，**--aaaa-rec** 选项用于指定 **A** 记录的当前值。使用 **--a-ip-address** 选项设置新值。 | **ipa dnsrecord-mod idm.example.com --aaaa-rec 2001:db8::1231:5675 --aaaa-ip-address 2001:db8::1231:5676** |

**表 99.4. "PTR"记录选项**

| 选项 | 描述 | 示例 |
| --- | --- | --- |
| **--ptr-rec**=*PTRRECORD* | 传递单个 PTR 记录或 PTR 记录列表。添加反向 DNS 记录时，与添加其他 DNS 记录的用法不同，与 **ipa dnsrecord-add** 命令一起使用的区域名称相反。通常，主机 IP 地址是给定网络中 IP 地址的最后一个八进制数。右侧的第一个示例为 server4.idm.example.com 添加 PTR 记录，其 IPv4 地址 为 192.168.122.4。第二个示例在 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa **中添加一个反向 DNS 条目。** 主机 server2.example.com 的 IPv6 反向区域，IP 地址为 2001:DB8::1111. | **ipa dnsrecord-add 122.168.192.in-addr.arpa 4 --ptr-rec server4.idm.example.com.**<br>**$ ipa dnsrecord-add 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.1.1.1.0.0.0.0.0.0.0.0.0.0.0 --ptr-rec server2.idm.example.com.** |
| **--ptr-hostname**=*string* | 提供记录的主机名。 | |

**表 99.5. "SRV"记录选项**

| 选项 | 描述 | 示例 |
| --- | --- | --- |
| **--srv-rec**=*SRVRECORD* | 传递单个 SRV 记录或 SRV 记录列表。在右侧的示例中，_ ldap._tcp 定义 SRV 记录的服务类型和连接协议。**srv-rec** 选项定义优先级、权重、端口和目标值。示例中的权重值为 51 和 49（总和为 100），它们代表使用特定记录的可能性（以百分比表示）。 | **# ipa dnsrecord-add idm.example.com _ldap._tcp --srv-rec="0 51 389 server1.idm.example.com."**<br>**# IPA dnsrecord-add server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com."** |

| *选项* | *描述* | *示例* |
| --- | --- | --- |
| **--srv-priority**=*number* | 设置记录的优先级。某一服务类型可以有多个 SRV 记录。优先级(0 - 65535)设置记录的排名；数值越低，优先级越高。服务必须首先使用优先级最高的记录。 | **# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com." --srv-priority=0** |
| **--srv-weight**=*number* | 设置记录的权重。这有助于确定优先级相同的 SRV 记录的顺序。集合权重应加到 100，代表使用特定记录的可能性（百分比）。 | **# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 49 389 server2.idm.example.com." --srv-weight=60** |
| **--srv-port**=*number* | 指定目标主机上 服务的端口。 | **# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 60 389 server2.idm.example.com." --srv-port=636** |
| **--srv-target**=*string* | 指定目标主机的域名。如果域中的服务不可用，这可以是单个句点(.)。 | |

**其它资源**

- 运行 **ipa dnsrecord-add --help**。

## 99.3. 确保使用 ANSIBLE 在 IDM 中存在 A 和 AAAA DNS 记录

**按照以下流程，使用 Ansible playbook 确保特定 IdM 主机的 A 和 AAAA 记录存在。在下面的示例中，IdM 管理员确保 idm.example.com DNS 区域中存在 host1 的 A 和 AAAA 记录。**

**先决条件**

- **您已配置了 Ansible 控制节点以满足以下要求：**
  - **您使用 Ansible 版本 2.14 或更高版本。**
  - **您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。**
  -

示例假定在 *~/MyPlaybooks/* 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 您知道 IdM 管理员密码。

- idm.example.com 区域存在，并由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅使用 Ansible playbook 管理 IdM DNS 区域。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 ensure-A- and-AAAA-records-are-present.yml Ansible playbook 文件的副本。例如：

```
$ cp ensure-A-and-AAAA-records-are-present.yml ensure-A-and-AAAA-records-are-present-copy.yml
```

4. 打开 ensure-A-and-AAAA-records-are-present-copy.yml 文件以进行编辑。

5. 通过在 ipadnsrecord 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 zone_name 变量设置为 idm.example.com。
- 在 records 变量中，将 name 变量设置为 host1，并将 a_ip_address 变量设置为 192.168.122.123。
- 在 records 变量中，将 name 变量设置为 host1，并将 aaaa_ip_address 变量设置为 ::1。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Ensure A and AAAA records are present
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure A and AAAA records are present
  - name: Ensure that 'host1' has A and AAAA records.
    ipadnsrecord:
      ipaadmin_password: "{{ ipaadmin_password }}"
      zone_name: idm.example.com
      records:
      - name: host1
        a_ip_address: 192.168.122.123
      - name: host1
        aaaa_ip_address: ::1
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-A-and-AAAA-records-are-present-copy.yml
```

**其它资源**

- 请参阅 IdM 中的 DNS 记录。

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsrecord.md 文件。

- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录中的 Ansible playbook 示例。

### 99.4. 确保使用 ANSIBLE 在 IDM 中存在 A 和 PTR DNS 记录

按照以下流程，使用 Ansible playbook 确保特定 IdM 主机的 A 记录存在，且包含对应的 PTR 记录。在以下流程中使用的示例中，IdM 管理员确保在 idm.example.com 区域中存在 IP 地址为 192.168.122.45 的 host1 的 A 和 PTR 记录。

#### 先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。
- idm.example.com DNS 区域存在，并由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅使用 Ansible playbook 管理 IdM DNS 区域。

#### 流程

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 ensure-dnsrecord-with-reverse-is-present.yml Ansible playbook 文件的副本。例如：

```
$ cp ensure-dnsrecord-with-reverse-is-present.yml ensure-dnsrecord-with-reverse-is-present-copy.yml
```

4. 打开 ensure-dnsrecord-with-reverse-is-present-copy.yml 文件以进行编辑。

5. 通过在 ipadnsrecord 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 name 变量设置为 host1。
- 将 zone_name 变量设置为 idm.example.com。
- 将 ip_address 变量设置为 192.168.122.45。
- 将 create_reverse 变量设置为 yes。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Ensure DNS Record is present.
```

```
hosts: ipaserver
become: true
gather_facts: false

tasks:
# Ensure that dns record is present
- ipadnsrecord:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: host1
    zone_name: idm.example.com
    ip_address: 192.168.122.45
    create_reverse: yes
    state: present
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-dnsrecord-with-reverse-is-present-copy.yml
```

**其它资源**

- 请参阅 IdM 中的 DNS 记录。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsrecord.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录中的 Ansible playbook 示例。

## 99.5. 确保使用 ANSIBLE 在 IDM 中存在多个 DNS 记录

按照以下流程，使用 Ansible playbook 确保多个值与特定 IdM DNS 记录相关联。在以下流程中使用的示例中，IdM 管理员确保 idm.example.com DNS 区域中存在 host1 的多个 A 记录。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：

- 您使用 Ansible 版本 2.14 或更高版本。
- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。
- idm.example.com 区域存在，并由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅使用 Ansible playbook 管理 IdM DNS 区域。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 ensure-presence-multiple-records.yml Ansible playbook 文件的副本。例如：

```
$ cp ensure-presence-multiple-records.yml ensure-presence-multiple-records-copy.yml
```

4. 打开 ensure-presence-multiple-records-copy.yml 文件进行编辑。

5. 通过在 ipadnsrecord 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 在 records 部分中，将 name 变量设置为 host1。
- 在 records 部分中，将 zone_name 变量设置为 idm.example.com。
- 在 records 部分中，将 a_rec 变量设置为 192.168.122.112，并将 192.168.122.1 22 设为 192.168.122.122。
- 在 record 部分中定义第二个 记录 ：
  - 将 name 变量设置为 host1。
  - 将 zone_name 变量设置为 idm.example.com。
  - 将 aaaa_rec 变量设置为 ::1。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Test multiple DNS Records are present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure that multiple dns records are present
  - ipadnsrecord:
      ipaadmin_password: "{{ ipaadmin_password }}"
      records:
        - name: host1
```

```
    zone_name: idm.example.com
    a_rec: 192.168.122.112
    a_rec: 192.168.122.122
  - name: host1
    zone_name: idm.example.com
    aaaa_rec: ::1
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-multiple-records-copy.yml
```

**其它资源**

- 请参阅 IdM 中的 DNS 记录。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsrecord.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录中的 Ansible playbook 示例。

## 99.6. 确保使用 ANSIBLE 在 IDM 中存在多个 CNAME 记录

Canonical Name 记录（CNAME 记录）是在域名系统(DNS)中一种资源记录类型，用于将一个域名、别名映射到另一个名称，即规范名称。

从单个 IP 地址运行多个服务时，您可能会发现 CNAME 记录很有用：例如，FTP 服务和 Web 服务，每个服务在不同端口上运行。

按照以下流程，使用 Ansible playbook 确保多个 CNAME 记录在 IdM DNS 中存在。在以下步骤中使用的示例中，host03 同时是 HTTP 服务器和 FTP 服务器。IdM 管理员确保在 idm.example.com 区域中存在 host03 A 记录的 www 和 ftp CNAME 记录。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。
- idm.example.com 区域存在，并由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅使用 Ansible playbook 管理 IdM DNS 区域。
- host03 A 记录存在于 idm.example.com 区域中。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 ensure-CNAME-record-is-present.yml Ansible playbook 文件的副本。例如：

```
$ cp ensure-CNAME-record-is-present.yml ensure-CNAME-record-is-present-copy.yml
```

4. 打开 ensure-CNAME-record-is-present-copy.yml 文件进行编辑。

5. 通过在 ipadnsrecord 任务部分设置以下变量来调整文件：

- （可选）调整 play 名称 提供的描述。
- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 zone_name 变量设置为 idm.example.com。
- 在 Record 变量 部分中，设置以下变量和值：
  - 将 name 变量设置为 www。
  - 将 cname_hostname 变量设置为 host03。
  - 将 name 变量设置为 ftp。
  - 将 cname_hostname 变量设置为 host03。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Ensure that 'www.idm.example.com' and 'ftp.idm.example.com' CNAME
records point to 'host03.idm.example.com'.
  hosts: ipaserver
  become: true
  gather_facts: false
```

```
tasks:
- ipadnsrecord:
    ipaadmin_password: "{{ ipaadmin_password }}"
    zone_name: idm.example.com
    records:
    - name: www
      cname_hostname: host03
    - name: ftp
      cname_hostname: host03
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-CNAME-record-is-present.yml
```

**其它资源**

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsrecord.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录中的 Ansible playbook 示例。

## 99.7. 使用 ANSIBLE 在 IDM 中存在 SRV 记录

DNS 服务 (SRV) 记录定义域中可用服务的主机名、端口号、传输协议、优先级和权重。在 Identity Management(IdM)中，您可以使用 SRV 记录来定位 IdM 服务器和副本。

按照以下流程，使用 Ansible playbook 确保 SRV 记录在 IdM DNS 中存在。在以下示例中，IdM 管理员可确保存在 _kerberos._udp.idm.example.com SRV 记录，其值为 10 50 88 idm.example.com。这会设置以下值：

- 它将服务的优先级设置为 10。
- 它将服务的权重设置为 50。

- 它将服务要使用的端口设置为 88。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
- 您知道 IdM 管理员密码。
- idm.example.com 区域存在，并由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅使用 Ansible playbook 管理 IdM DNS 区域。

**流程**

1. 导航到 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开清单文件，并确保 [ipaserver] 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 server.idm.example.com，请输入：

```
[ipaserver]
server.idm.example.com
```

3. 生成 ensure-SRV-record-is-present.yml Ansible playbook 文件的副本。例如：

```
$ cp ensure-SRV-record-is-present.yml ensure-SRV-record-is-present-copy.yml
```

4. 打开 ensure-SRV-record-is-present-copy.yml 文件进行编辑。

5. 通过在 ipadnsrecord 任务部分设置以下变量来调整文件：

- 将 ipaadmin_password 变量设置为 IdM 管理员密码。
- 将 name 变量设置为 _kerberos._udp.idm.example.com。
- 将 srv_rec 变量设置为 '10 50 88 idm.example.com'。
- 将 zone_name 变量设置为 idm.example.com。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- name: Test multiple DNS Records are present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure a SRV record is present
  - ipadnsrecord:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: _kerberos._udp.idm.example.com
      srv_rec: ’10 50 88 idm.example.com’
      zone_name: idm.example.com
      state: present
```

6. 保存该文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-SRV-record-is-present.yml
```

**其它资源**

- 请参阅 IdM 中的 DNS 记录。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-dnsrecord.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 目录中的 Ansible playbook 示例。

# 第 100 章 使用 ANSIBLE 管理 IDM 服务器

您可以使用 Red Hat Ansible Engine 来管理身份管理(IdM)拓扑中的服务器。您可以使用 ansible-freeipa 软件包中的 server 模块来检查 IdM 拓扑中是否存在服务器。您还可以隐藏任何副本或使副本可见。

这部分包含以下主题：

- 使用 Ansible 检查 IdM 服务器是否存在
- 使用 Ansible 确保 IdM 拓扑中没有 IdM 服务器
- 确保尽管拥有最后一个 IdM 服务器角色，也不存在 IdM 服务器
- 确保 IdM 服务器不存在，但不一定与其他 IdM 服务器断开连接
- 使用 Ansible playbook 确保现有的 IdM 服务器被隐藏
- 使用 Ansible playbook 确保现有的 IdM 服务器可见
- 确保现有的 IdM 服务器被分配了 IdM DNS 位置
- 确保现有的 IdM 服务器没有分配 IdM DNS 位置

## 100.1. 使用 ANSIBLE 检查 IDM 服务器是否存在

您可以在 Ansible playbook 中使用 ipaserver ansible-freeipa 模块来验证是否存在身份管理(IdM)服务器。

**注意**

ipaserver Ansible 模块不会安装 IdM 服务器。

**先决条件**

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
  - 清单文件中定义的从控制节点到 IdM 服务器的 SSH 连接工作正常。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/server/ 目录中的 server-present.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-present.yml server-present-copy.yml
```

3. 打开 server-present-copy.yml 文件进行编辑。

4. 通过在 ipaserver 任务部分中设置以下变量来调整文件，并保存文件：

- 将 ipaadmin_password 变量设为 IdM admin 的密码。
- 将 name 变量设为服务器的 FQDN。示例服务器的 FQDN 是 server123.idm.example.com。

```
---
- name: Server present example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com is present
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
```

5. 运行 Ansible playbook，并指定 playbook 文件和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-present-copy.yml
```

**其它资源**

- 请参阅 使用 Ansible playbook 安装身份管理服务器。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-server.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/server 目录中的 playbook 示例。

## 100.2. 使用 ANSIBLE 确保 IDM 拓扑中没有 IDM 服务器

使用 Ansible playbook 确保 IdM 拓扑中不存在身份管理(IdM)服务器，即使作为主机也不存在。

与 ansible-freeipa ipaserver 角色不同，此 playbook 中使用的 ipaserver 模块不会从服务器卸载 IdM 服务。

**先决条件**

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
  - 清单文件中定义的从控制节点到 IdM 服务器的 SSH 连接工作正常。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/server/ 目录中的 server-absent.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-absent.yml server-absent-copy.yml
```

3. 打开 server-absent-copy.yml 文件进行编辑。

4. 通过在 ipaserver 任务部分中设置以下变量来调整文件，并保存文件：

- 将 ipaadmin_password 变量设为 IdM admin 的密码。
- 将 name 变量设为服务器的 FQDN。示例服务器的 FQDN 是 server123.idm.example.com。
- 确保 state 变量设置为 absent。

```
---
- name: Server absent example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com is absent
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      state: absent
```

5. 运行 Ansible playbook，并指定 playbook 文件和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-absent-copy.yml
```

6. 确保指向 server123.idm.example.com 的所有名称服务器(NS)DNS 记录都已从 DNS 区域中删除。无论您使用由 IdM 还是外部 DNS 管理的集成 DNS，这个均适用。

**其它资源**

- 请参阅卸载 IdM 服务器。

- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-server.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/server 目录中的 playbook 示例。

### 100.3. 确保尽管拥有最后一个 IDM 服务器角色，也不存在 IDM 服务器

您可以使用 Ansible 来确保没有身份管理(IdM)服务器，即使最后一个 IdM 服务实例正在服务器上运行。证书颁发机构(CA)、密钥恢复机构(KRA)或 DNS 服务器都是 IdM 服务的示例。

**警告**

如果您删除了作为 CA、KRA 或 DNS 服务器的最后一台服务器，会严重破坏 IdM 功能。您可以使用 ipa service-find 命令手动检查哪些服务运行在哪些 IdM 服务器上。CA 服务器的主要名称为 dogtag/*server_name*/*REALM_NAME*。

与 ansible-freeipa ipaserver 角色不同，此 playbook 中使用的 ipaserver 模块不会从服务器卸载 IdM 服务。

**先决条件**

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
  - 清单文件中定义的从控制节点到 IdM 服务器的 SSH 连接工作正常。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/server/ 目录中的 server-absent-ignore-last-of-role.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-absent-ignore-last-of-role.yml server-absent-ignore-last-of-role-copy.yml
```

3. 打开 server-absent-ignore-last-of-role-copy.yml 文件进行编辑。

4. 通过在 ipaserver 任务部分中设置以下变量来调整文件，并保存文件：
   - 将 ipaadmin_password 变量设为 IdM admin 的密码。
   - 将 name 变量设为服务器的 FQDN。示例服务器的 FQDN 是 server123.idm.example.com。
   - 确保 ignore_last_of_role 变量设为 yes。
   - 将 state 变量设置为 absent。

```
---
- name: Server absent with last of role skip example
```

```
hosts: ipaserver
vars_files:
- /home/user_name/MyPlaybooks/secret.yml
tasks:
- name: Ensure server “server123.idm.example.com” is absent with last of role skip
  ipaserver:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: server123.idm.example.com
    ignore_last_of_role: yes
    state: absent
```

5. 运行 Ansible playbook，并指定 playbook 文件和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-absent-ignore-last-of-role-copy.yml
```

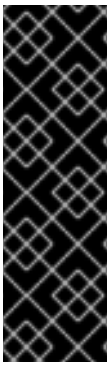
6. 确保指向 server123.idm.example.com 的所有名称服务器(NS)DNS 记录已从 DNS 区域中删除。无论您使用由 IdM 还是外部 DNS 管理的集成 DNS，这个均适用。

**其它资源**

- 请参阅卸载 IdM 服务器。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-server.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/server 目录中的 playbook 示例。

## 100.4. 确保 IDM 服务器不存在，但不一定与其他 IDM 服务器断开连接

如果要从拓扑中删除身份管理(IdM)服务器，您可以使用 Ansible playbook 使其复制协议保持不变。playbook 还确保 IdM 服务器在 IdM 中不存在，即使作为主机也是如此。



**重要**

仅当其他服务器是您计划删除的工作不正常的服务器时，才建议在删除时忽略服务器的复制协议。删除拓扑中作为中心点的服务器会将拓扑分成两个断开连接的集群。

您可以使用 ipa server-del 命令从拓扑中删除工作不正常的服务器。

**注意**

如果删除了作为证书颁发机构(CA)、密钥恢复机构(KRA)或 DNS 服务器的最后一台服务器，将会严重破坏身份管理(IdM)功能。为防止此问题，playbook 在卸载充当 CA、KRA 或 DNS 服务器的服务器之前，确保这些服务运行在域中的另一台服务器上。

与 ansible-freeipa ipaserver 角色不同，此 playbook 中使用的 ipaserver 模块不会从服务器卸载 IdM 服务。

**先决条件**

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
  - 清单文件中定义的从控制节点到 IdM 服务器的 SSH 连接工作正常。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/server/ 目录中的 server-absent-ignore_topology_disconnect.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-absent-ignore_topology_disconnect.yml server-absent-ignore_topology_disconnect-copy.yml
```

3. 打开 server-absent-ignore_topology_disconnect-copy.yml 文件进行编辑。

4. 通过在 ipaserver 任务部分中设置以下变量来调整文件，并保存文件：

- 将 ipaadmin_password 变量设为 IdM admin 的密码。
- 将 name 变量设为服务器的 FQDN。示例服务器的 FQDN 是 server123.idm.example.com。
- 确保 ignore_topology_disconnect 变量设为 yes。
- 确保 state 变量设置为 absent。

```
---
- name: Server absent with ignoring topology disconnects example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server “server123.idm.example.com” with ignoring topology disconnects
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      ignore_topology_disconnect: yes
      state: absent
```

5. 运行 Ansible playbook，并指定 playbook 文件和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-absent-ignore_topology_disconnect-copy.yml
```

6.

[可选] 确保指向 server123.idm.example.com 的所有名称服务器(NS)DNS 记录已从 DNS 区域中删除。无论您使用由 IdM 还是外部 DNS 管理的集成 DNS，这个均适用。

其它资源

- 请参阅卸载 IdM 服务器。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-server.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/server 目录中的 playbook 示例。

### 100.5. 使用 ANSIBLE PLAYBOOK 确保现有的 IDM 服务器被隐藏

使用 Ansible playbook 中的 ipaserver ansible-freeipa 模块，来确保隐藏现有的身份管理(IdM)服务器被隐藏了。请注意，此 playbook 没有安装 IdM 服务器。

先决条件

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 清单文件中定义的从控制节点到 IdM 服务器的 SSH 连接工作正常。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/server/ 目录中的 server-hidden.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-hidden.yml server-hidden-copy.yml
```

3. 打开 server-hidden-copy.yml 文件进行编辑。

4. 通过在 ipaserver 任务部分中设置以下变量来调整文件，并保存文件：

- 将 ipaadmin_password 变量设为 IdM admin 的密码。
- 将 name 变量设为服务器的 FQDN。示例服务器的 FQDN 是 server123.idm.example.com。
- 确保 hidden 变量设为 True。

```
---
- name: Server hidden example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com is hidden
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      hidden: True
```

5. 运行 Ansible playbook，并指定 playbook 文件和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-hidden-copy.yml
```

其它资源

- 请参阅 使用 Ansible playbook 安装身份管理服务器。
- 请参阅 隐藏的副本模式。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-server.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/server 目录中的 playbook 示例。

## 100.6. 使用 ANSIBLE PLAYBOOK 确保现有的 IDM 服务器可见

使用 Ansible playbook 中的 ipaserver ansible-freeipa 模块，来确保可以现有的身份管理(IdM)服务器可见。请注意，此 playbook 没有安装 IdM 服务器。

先决条件

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。

- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 清单文件中定义的从控制节点到 IdM 服务器的 SSH 连接工作正常。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/server/ 目录中的 server-not-hidden.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-not-hidden.yml server-not-hidden-copy.yml
```

3. 打开 server-not-hidden-copy.yml 文件进行编辑。

4. 通过在 ipaserver 任务部分中设置以下变量来调整文件，并保存文件：

- 将 ipaadmin_password 变量设为 IdM admin 的密码。

- 将 name 变量设为服务器的 FQDN。示例服务器的 FQDN 是 server123.idm.example.com。

- 确保 hidden 变量设为 no。

```
---
- name: Server not hidden example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
```

```
- name: Ensure server server123.idm.example.com is not hidden
  ipaserver:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: server123.idm.example.com
    hidden: no
```

5. 运行 Ansible playbook，并指定 playbook 文件和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-not-hidden-copy.yml
```

**其它资源**

- 请参阅 使用 Ansible playbook 安装身份管理服务器。
- 请参阅 隐藏的副本模式。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-server.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/server 目录中的 playbook 示例。

### 100.7. 确保现有的 IDM 服务器被分配了 IDM DNS 位置

使用 Ansible playbook 中的 ipaserver ansible-freeipa 模块来确保为现有身份管理(IdM)服务器分配了特定的 IdM DNS 位置。

请注意，ipaserver Ansible 模块没有安装 IdM 服务器。

**先决条件**

- 您需要知道 IdM admin 密码。
- IdM DNS 位置存在。位置示例为 germany。
- 您有访问服务器的 root 权限。服务器示例是 server123.idm.example.com。

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
  - 清单文件中定义的从控制节点到 IdM 服务器的 SSH 连接工作正常。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/server/ 目录中的 server-location.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-location.yml server-location-copy.yml
```

3. 打开 server-location-copy.yml 文件进行编辑。

4. 通过在 ipaserver 任务部分中设置以下变量来调整文件，并保存文件：

- 将 ipaadmin_password 变量设为 IdM admin 的密码。
- 将 name 变量设为 server123.idm.example.com。
- 将 location 变量设为 germany。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Server enabled example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com with location “germany” is
present
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      location: germany
```

5. 运行 Ansible playbook，并指定 playbook 文件和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-location-
copy.yml
```

6. 以 root 用户身份使用 SSH 连接到 server123.idm.example.com ：

```
ssh root@server123.idm.example.com
```

7. 重新启动服务器上的 named-pkcs11 服务，以使更新立即生效：

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

**其它资源**

- 请参阅 使用 Ansible playbook 安装身份管理服务器。

- 请参阅 使用 Ansible 来确保 IdM 位置存在。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-server.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/server 目录中的 playbook 示例。

## 100.8. 确保现有的 IDM 服务器没有分配 IDM DNS 位置

使用 Ansible playbook 中的 ipaserver ansible-freeipa 模块，来确保现有身份管理(IdM)服务器没有为其分配的 IdM DNS 位置。不要将 DNS 位置分配给经常更改地理位置的服务器。请注意，playbook 不安装 IdM 服务器。

### 先决条件

- 您需要知道 IdM admin 密码。
- 您有访问服务器的 root 权限。服务器示例是 server123.idm.example.com。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

- 清单文件中定义的从控制节点到 IdM 服务器的 SSH 连接工作正常。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/server/ 目录中的 server-no-location.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-no-location.yml server-no-location-copy.yml
```

3. 打开 server-no-location-copy.yml 文件进行编辑。

4. 通过在 ipaserver 任务部分中设置以下变量来调整文件，并保存文件：

- 将 ipaadmin_password 变量设为 IdM admin 的密码。
- 将 name 变量设为 server123.idm.example.com。
- 确保 location 变量设为 ""。

```
---
- name: Server no location example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com is present with no location
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      location: “”
```

5. 运行 Ansible playbook，并指定 playbook 文件和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-no-location-copy.yml
```

6. 以 root 用户身份使用 SSH 连接到 server123.idm.example.com ：

```
ssh root@server123.idm.example.com
```

7. 重新启动服务器上的 named-pkcs11 服务，以使更新立即生效：

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

**其它资源**

- 请参阅 使用 Ansible playbook 安装身份管理服务器。
- 请参阅 在 IdM 中使用 Ansible 来管理 DNS 位置。
- 请参阅 /usr/share/doc/ansible-freeipa/ 目录中的 README-server.md 文件。
- 请参阅 /usr/share/doc/ansible-freeipa/playbooks/server 目录中的 playbook 示例。

# 第 101 章 收集 IDM 健康检查信息

健康检查已设计为手动命令行工具，可帮助您识别身份管理(IdM)中可能存在的问题。

您可以根据 30 天轮转的 Healthcheck 输出创建一个日志集合。

**先决条件**

- Healthcheck 工具仅适用于 RHEL 8.1 或更新版本

## 101.1. IDM 中的 HEALTHCHECK

身份管理(IdM)中的 Healthcheck 工具可帮助发现可能影响 IdM 环境健康的问题。

**注意**

Healthcheck 工具是一个命令行工具，可在无需 Kerberos 身份验证的情况下使用。

**模块是独立的**

Healthcheck由独立模块组成，用于测试：

- 复制问题
- 证书有效期
- 证书颁发机构基础设施问题
- IdM 和 Active Directory 信任问题
- 正确的文件权限和所有权设置

两种输出格式

HealthCheck 生成以下输出，您可以使用 output-type 选项来设置：

- JSON ：JSON 格式的机器可读输出（默认）
- human：人类可读的输出

您可以使用 --output-file 选项来指定不同的文件目标。

结果

每个 Healthcheck 模块返回以下结果之一：

SUCCESS

配置为预期

WARNING

不是错误，但需要对其进行检查和评估

ERROR

未按预期配置

CRITICAL

未按预期配置，可能会有非常大的影响

### 101.2. 日志轮转

日志轮转每日创建新的日志文件，并且按日期组织这些文件。由于日志文件保存在同一目录中，因此您可以根据日期选择特定的日志文件。

轮转意味着为最多日志文件数配置一个数字，如果超过这个数字，则最新文件重写并重命名最旧的文件。例如，如果轮转编号为 30，则第三十个日志文件将取代第一个（最旧的）日志文件。

日志轮转会减少大量日志文件并组织它们，这有助于分析日志。

### 101.3. 使用 IDM HEALTHCHECK 配置日志轮转

按照以下流程配置日志轮转：

- systemd 计时器
- crond 服务

systemd 定时器定期运行 Healthcheck 工具并生成日志。默认值设为每天的上午 4 点。

crond 服务用于日志轮转。

默认日志名称为 healthcheck.log，轮转的日志使用 healthcheck.log-YYYMMDD 格式。

**先决条件**

- 您必须以 root 用户身份执行命令。

**流程**

1. 启用 systemd 计时器：

```
# systemctl enable ipa-healthcheck.timer
Created symlink /etc/systemd/system/multi-user.target.wants/ipa-healthcheck.timer ->
/usr/lib/systemd/system/ipa-healthcheck.timer.
```

2. 启动 systemd 计时器：

```
# systemctl start ipa-healthcheck.timer
```

3. 打开 /etc/logrotate.d/ipahealthcheck 文件，以配置应保存的日志数。

默认情况下，日志轮转设置为 30 天。

4. 在 /etc/logrotate.d/ipahealthcheck 文件中，配置日志的路径。

   默认情况下，日志保存在 /var/log/ipa/healthcheck/ 目录中。

5. 在 /etc/logrotate.d/ipahealthcheck 文件中，配置日志生成时间。

   默认情况下，日志在每天的上午 4 点创建。

6. 要使用日志轮转，请确保 crond 服务已启用并在运行：

```
# systemctl enable crond
# systemctl start crond
```

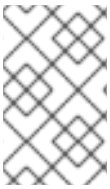
要开始生成日志，启动 IPA healthcheck 服务：

```
# systemctl start ipa-healthcheck
```

要验证结果，请转至 /var/log/ipa/healthcheck/，并检查日志是否已正确创建。

### 101.4. 更改 IDM HEALTHCHECK 配置

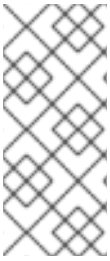
您可以通过在 /etc/ipahealthcheck/ipahealthcheck.conf 文件中添加所需的命令行选项来更改 Healthcheck 设置。这很有用，例如，您配置了日志轮转，并希望确保日志采用适合自动分析的格式，但不想设置新的计时器。



**注意**

此 Healthcheck 功能仅适用于 RHEL 8.7 或更新版本。

在修改后，Healthcheck 创建的所有日志遵循新的设置。这些设置也应用到健康检查的任何手动执行。



**注意**

手动运行 Healthcheck 时，配置文件中的设置优先于命令行中指定的选项。例如，如果在配置文件中将 output_type 设为 human，则在命令行上指定 json 不起作用。您用来在配置文件中指定的任何命令行选项都会正常应用。

**其它资源**

- 使用 IdM Healthcheck 配置日志轮转

### 101.5. 配置 HEALTHCHECK 以更改输出日志格式

按照以下流程，配置带有已设置了计时器的 Healthcheck。在本例中，您将配置 Healthcheck 以人类可读格式生成日志，并且包含成功结果而不是仅错误。

**先决条件**

- 您的系统正在运行 RHEL 8.7 或更高版本。
- 您有 root 特权。
- 您之前已在计时器中配置了日志轮转。

**流程**

1. 在文本编辑器中打开 /etc/ipahealthcheck/ipahealthcheck.conf 文件。
2. 将 options output_type=human 和 all=True 添加到 [default] 部分。
3. 保存并关闭该文件。

验证

1. 手动运行 Healthcheck:

```
# ipa-healthcheck
```

2. 进入 /var/log/ipa/healthcheck/，检查日志是否采用正确的格式。

**其它资源**

- 使用 IdM Healthcheck 配置日志轮转

# 第 102 章 使用 IDM HEALTHCHECK 检查服务

您可以使用 Healthcheck 工具监控身份管理(IdM)服务器使用的服务。

详情请查看 IdM 中的 Healthcheck。

**先决条件**

- Healthcheck 工具只在 RHEL 8.1 及更新版本中可用

## 102.1. SERVICES HEALTHCHECK 测试

Healthcheck 工具包括一个测试，用于检查是否任何 IdM 服务没有在运行。此测试很重要，因为未运行的服务会在其他测试中造成失败。因此，请先检查所有服务是否都在运行。然后您可以检查所有其他测试结果。

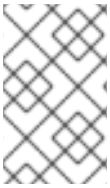
要查看所有服务测试，请运行 ipa-healthcheck 及 --list-sources选项：

```
# ipa-healthcheck --list-sources
```

您可以在 ipahealthcheck.meta.services 源下找到使用 Healthcheck 测试的所有服务：

- certmonger
- dirsrv
- gssproxy
- httpd
- ipa_custodia

- ipa_dnskeysyncd
- ipa_otpd
- kadmin
- krb5kdc
- named
- pki_tomcatd
- sssd



**注意**

当尝试发现问题时，在所有 IdM 服务器中运行这些测试。

## 102.2. 使用 HEALTHCHECK 的服务

按照以下流程，使用 Healthcheck 工具对在身份管理(IdM)服务器上运行的服务运行独立的手动测试。

Healthcheck 工具包括许多测试，其结果可通过以下方法缩短：

- 排除所有成功测试 ：--failures-only
- 仅包含服务测试 ：-- source=ipahealthcheck.meta.services

**流程**

- 要使用服务相关的警告、错误和严重问题运行健康检查，请输入：

```
# ipa-healthcheck --source=ipahealthcheck.meta.services --failures-only
```

成功测试会显示空括号：

```
[]
```

如果其中一个服务失败，则结果可能类似以下示例：

```
{
  "source": "ipahealthcheck.meta.services",
  "check": "httpd",
  "result": "ERROR",
  "kw": {
    "status": false,
    "msg": "httpd: not running"
  }
}
```

其它资源

- 请参阅 man ipa-healthcheck。

# 第 103 章 使用 IDM 健康检查验证您的 IDM 和 AD 信任配置

了解如何使用 Healthcheck 工具识别 IdM 和身份管理(IdM)中活动目录信任的问题。

**先决条件**

- Healthcheck 工具仅适用于 RHEL 8.1 或更新版本

## 103.1. IDM 和 AD 信任健康检查测试

Healthcheck 工具包括多个测试，用于测试您的身份管理(IdM)和 Active Directory(AD)信任状态。

要查看所有信任测试，请使用 **--list-sources** 选项运行 **ipa-healthcheck** ：

```
# ipa-healthcheck --list-sources
```

您可以在 **ipahealthcheck.ipa.trust** 源下找到所有测试：

**IPATrustAgentCheck**

当机器配置为信任代理时，这个测试会检查 SSSD 配置。对于 **/etc/sssd/sssd.conf** 中的每个域，其中 **id_provider=ipa** 确保 **ipa_server_mode** 为 **True**。

**IPATrustDomainsCheck**

此测试通过将 **sssctl domain-list** 中的域列表与 **ipa trust-find** 中排除了 IPA 域的域列表进行比较，来检查信任域是否与 SSSD 域匹配。

**IPATrustCatalogCheck**

此测试解析为 AD 用户 **Administrator@REALM**。这将填充 **sssctl domain-status** 输出中的 AD Global 目录和 AD 域控制器值。

对于每个信任域，查找 SID + 500（管理员） ID 的用户，然后检查 **sssctl domain-status <domain> --active-server** 的输出以确保域处于活跃状态。

**IPAsidgenpluginCheck**

此测试会验证 IPA 389-ds 实例中是否启用了 sidgen 插件。该测试还验证 **cn=plugins,cn=config** 中的 IPA SIDGEN 和 ipa-sidgen-task 插件是否包含 nsslapd-

pluginEnabled 选项。

IPATrustAgentMemberCheck

此测试将验证当前主机是否为 cn=adtrust 代理,cn=sysaccounts,cn=etc,SUFFIX 的成员。

IPATrustControllerPrincipalCheck

此测试将验证当前主机是否为 cn=adtrust 代理,cn=sysaccounts,cn=etc,SUFFIX 的成员。

IPATrustControllerServiceCheck

此测试会验证当前主机是否在 ipactl 中启动 ADTRUST 服务。

IPATrustControllerConfCheck

此测试验证 net conf 列表输出中是否为 passdb 后端启用了 ldapi。

IPATrustControllerGroupSIDCheck

此测试将验证 admin 组的 SID 是否以 512(Domain Admins RID)结束。

IPATrustPackageCheck

如果没有启用信任控制器和 AD 信任，这个测试会验证是否安装了 trust-ad 软件包。

注意

当尝试找到问题时，在所有 IdM 服务器中运行这些测试。

## 103.2. 使用 HEALTHCHECK 工具建立信任

按照以下流程，使用 Healthcheck 工具对身份管理(IdM)和活动目录(AD)信任健康检查运行独立的手动测试。

因此，Healthcheck 工具包含许多测试，您可以通过以下方式缩短结果：

- 排除所有成功测试 ：--failures-only
- 仅包含信任测试 ：-- source=ipahealthcheck.ipa.trust

**流程**

- 要运行带有信任中的警告、错误和严重问题的健康检查，请输入：

```
# ipa-healthcheck --source=ipahealthcheck.ipa.trust --failures-only
```

成功测试会显示空括号：

```
# ipa-healthcheck --source=ipahealthcheck.ipa.trust --failures-only
[]
```

**其它资源**

- 请参阅 man ipa-healthcheck。

# 第 104 章 使用 IDM HEALTHCHECK 验证证书

了解更多有关理解和使用身份管理(IdM)中的 Healthcheck 工具，以识别由 certmonger 维护的 IPA 证书的问题。

详情请查看 IdM 中的 Healthcheck。

**先决条件**

- Healthcheck 工具只在 RHEL 8.1 及更新版本中可用。

## 104.1. IDM 证书健康检查测试

Healthcheck 工具包括多个测试，用于验证 Identity Management(IdM)中由 certmonger 维护的证书状态。有关 certmonger 的详情，请参阅使用 certmonger 为服务获取 IdM 证书。

此测试套件检查过期、验证、信任和其他问题。对于相同的根本问题，可能会抛出多个错误。

要查看所有证书测试，请使用 **--list-sources** 选项运行 **ipa-healthcheck** ：

```
# ipa-healthcheck --list-sources
```

您可以在 ipahealthcheck.ipa.certs 源下找到所有测试：

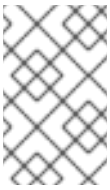
**IPACertmongerExpirationCheck**

此测试检查 certmonger 中的过期时间。

如果报告错误，证书已过期。

如果出现警告，证书很快就会过期。默认情况下，此测试在证书过期前 28 天或少于 28 天内适用。

您可以在 /etc/ipahealthcheck/ipahealthcheck.conf 文件中配置天数。打开该文件后，更改 default 部分中的 cert_expiration_days 选项。



**注意**

Certmonger 加载和维护自己的证书过期视图。此检查不会验证磁盘中的证书。

**IPACertfileExpirationCheck**

此测试检查证书文件或 NSS 数据库是否无法打开。此测试还会检查过期情况。因此，请仔细阅读错误或警告输出中的 msg 属性。消息指定了问题。

**注意**

此测试会检查磁盘中的证书。如果证书丢失、不可读取等单独错误，也可以引发单独的错误。

**IPACertNSSTrust**

此测试比较存储在 NSS 数据库中的证书的信任。对于 NSS 数据库中的预期跟踪证书，会将信任与预期值进行比较，并在不匹配时引发错误。

**IPANSSChainValidation**

此测试会验证 NSS 证书的证书链。测试执行： certutil -V -u V -e -d [dbdir] -n [nickname]

**IPAOpenSSLChainValidation**

此测试会验证 OpenSSL 证书的证书链。与 NSSChain 验证相当的 OpenSSL 命令是我们执行的 OpenSSL 命令：

```
openssl verify -verbose -show_chain -CAfile /etc/ipa/ca.crt [cert file]
```

**IPARAAgent**

此测试将磁盘上的证书与 LDAP in uid=ipara,ou=People,o=ipaca 中的等效记录进行比较。

**IPACertRevocation**

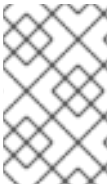
此测试使用 certmonger 验证证书是否已被撤销。因此，测试只能查找与 certmonger 维护的证书连接的问题。

**IPACertmongerCA**

此测试将验证证书授权机构(CA)配置。IdM 无法在没有 CA 的情况下发布证书。

Certmonger 维护一组 CA 帮助程序。在 IdM 中，有一个名为 IPA 的 CA，它通过 IdM 发布证书，它作为主机或用户主体进行身份验证，用于主机或服务证书。

还有一个 dogtag-ipa-ca-renew-agent 和 dogtag-ipa-ca-renew-agent-reuse （续订 CA 子系统证书）



**注意**

当尝试检查问题时，在所有 IdM 服务器中运行这些测试。

### 104.2. 使用 HEALTHCHECK 工具验证证书

按照以下流程，使用 Healthcheck 工具运行身份管理(IdM)证书健康检查的独立的手动测试。

因此，Healthcheck 工具包括了许多测试，您可以使用以下方法缩短结果：

- 排除所有成功测试 ：--failures-only
- 仅包含证书测试 ：-- source=ipahealthcheck.ipa.certs

**先决条件**

- 您必须以 root 用户身份执行 Healthcheck 测试。

**流程**

- 要使用证书的警告、错误和严重问题运行健康检查，请输入：

```
# ipa-healthcheck --source=ipahealthcheck.ipa.certs --failures-only
```

成功测试会显示空括号：

```
[]
```

失败的测试会显示以下输出：

```
{
  "source": "ipahealthcheck.ipa.certs",
  "check": "IPACertfileExpirationCheck",
  "result": "ERROR",
  "kw": {
    "key": 1234,
    "dbdir": "/path/to/nssdb",
    "error": [error],
    "msg": "Unable to open NSS database '/path/to/nssdb': [error]"
  }
}
```

在打开 NSS 数据库时，这个 IPACertfileExpirationCheck 测试失败。

其它资源

- 请参阅 man ipa-healthcheck。

# 第 105 章 使用 IDM HEALTHCHECK 验证系统证书

了解如何使用 Healthcheck 工具识别身份管理(IdM)中系统证书的问题。

详情请查看 IdM 中的 Healthcheck。

**先决条件**

- Healthcheck 工具仅在 RHEL 8.1 或更新版本中可用。

## 105.1. 系统证书健康检查测试

Healthcheck 工具包括一些用于验证系统(DogTag)证书的测试。

要查看所有测试，请使用 --list-sources 选项运行 ipa-healthcheck：

```
# ipa-healthcheck --list-sources
```

您可以在 ipahealthcheck.dogtag.ca 源下找到所有测试：

**DogtagCertsConfigCheck**

此测试将其 NSS 数据库中的 CA（证书授权机构）证书与存储在 CS.cfg 中的相同值进行比较。如果不匹配，CA 无法启动。

具体来说，它会检查：

- auditSigningCert cert-pki-ca against ca.audit_signing.cert
- ocspSigningCert cert-pki-ca against ca.ocsp_signing.cert
- caSigningCert cert-pki-ca against ca.signing.cert

- subsystemCert cert-pki-ca against ca.subsystem.cert
- 针对 ca.sslserver.cert 的 Server-Cert cert-pki-ca

如果安装了 Key Recovery Authority(KRA) :

- transportCert cert-pki-kra against ca.connector.KRA.transportCert

DogtagCertsConnectivityCheck

此测试验证连接性。这个测试等同于检查的 ipa cert-show 1 命令 :

- Apache 中的 PKI 代理配置
- IdM 能够找到 CA
- RA 代理客户端证书
- CA 回复请求的正确性

请注意，测试会使用 serial #1 检查证书，因为您要验证是否 可以执行证书 并返回 CA 中的预期结果（证书或未找到）。

注意

当尝试找到问题时，在所有 IdM 服务器中运行这些测试。

## 105.2. 使用 HEALTHCHECK 强制系统证书

按照以下流程，使用 Healthcheck 工具运行身份管理(IdM)证书的独立的手动测试。

由于 Healthcheck 工具包含许多测试，因此您可以通过仅包含 DogTag 测试来缩小结果范围：--source=ipahealthcheck.dogtag.ca

流程

- 要运行限制为 DogTag 证书的 Healthcheck，请输入：

```
# ipa-healthcheck --source=ipahealthcheck.dogtag.ca
```

测试成功示例：

```
{
  "source: ipahealthcheck.dogtag.ca",
  "check: DogtagCertsConfigCheck",
  "result: SUCCESS",
  "uuid: 9b366200-9ec8-4bd9-bb5e-9a280c803a9c",
  "when: 20191008135826Z",
  "duration: 0.252280",
  "kw:" {
    "key": "Server-Cert cert-pki-ca",
    "configfile":  "/var/lib/pki/pki-tomcat/conf/ca/CS.cfg"
    }
}
```

测试失败的示例：

```
{
  "source: ipahealthcheck.dogtag.ca",
  "check: DogtagCertsConfigCheck",
  "result: CRITICAL",
  "uuid: 59d66200-1447-4b3b-be01-89810c803a98",
  "when: 20191008135912Z",
  "duration: 0.002022",
  "kw:" {
    "exception": "NSDB /etc/pki/pki-tomcat/alias not initialized",
    }
}
```

其它资源

- 请参阅 man ipa-healthcheck。

# 第 106 章 使用 IDM HEALTHCHECK 检查磁盘空间

您可以使用 Healthcheck 工具监控身份管理服务器的可用磁盘空间。

详情请查看 IdM 中的 Healthcheck。

**先决条件**

- Healthcheck 工具仅适用于 RHEL 8.1 及更新的版本。

## 106.1. 磁盘空间健康检查测试

Healthcheck 工具包括用于检查可用磁盘空间的测试。可用磁盘空间不足可能会导致以下问题：

- 日志
- 执行
- Backups

测试检查以下路径：

**表 106.1. 测试的路径**

| 测试检查的路径 | 以 MB 为单位的最小磁盘空间 |
|---|---|
| /var/lib/dirsrv/ | 1024 |
| /var/lib/ipa/backup/ | 512 |
| /var/log/ | 1024 |
| var/log/audit/ | 512 |
| /var/tmp/ | 512 |

| 测试检查的路径 | 以 MB 为单位的最小磁盘空间 |
| --- | --- |
| /tmp/ | 512 |

要列出所有测试，请使用 --list-sources 选项运行 ipa-healthcheck：

```
# ipa-healthcheck --list-sources
```

您可以在 ipahealthcheck.system.filesystemspace 源中找到文件系统空间检查测试：

FileSystemSpaceCheck

此测试以以下方式检查可用磁盘空间：

- 需要最少的原始可用字节数。
- 最小可用磁盘空间百分比为 20%。

## 106.2. 使用 HEALTHCHECK 工具强制磁盘空间

按照以下流程，使用 Healthcheck 工具在身份管理(IdM)服务器上运行可用磁盘空间的独立的手动测试。

因为健康检查包括许多测试，因此您可以通过以下方式缩小结果范围：

- 排除所有成功测试 ：--failures-only
- 仅包含空间检查测试 ：-- source=ipahealthcheck.system.filesystemspace

流程

- 要使用可用磁盘空间的警告、错误和严重问题运行健康检查，请输入：

```
# ipa-healthcheck --source=ipahealthcheck.system.filesystemspace --failures-only
```

成功测试会显示空括号：

```
[]
```

例如，测试失败可显示：

```
{
  "source": "ipahealthcheck.system.filesystemspace",
  "check": "FileSystemSpaceCheck",
  "result": "ERROR",
  "kw": {
    "msg": "/var/lib/dirsrv: free space under threshold: 0 MiB < 1024 MiB",
    "store": "/var/lib/dirsrv",
    "free_space": 0,
    "threshold": 1024
  }
}
```

失败的测试会通知您 /var/lib/dirsrv 目录已用尽空间。

其它资源

- 请参阅 man ipa-healthcheck。

# 第 107 章 使用 HEALTHCHECK 验证 IDM 配置文件的权限

了解如何使用 Healthcheck 工具测试身份管理(IdM)配置文件。

详情请查看 IdM 中的 Healthcheck。

**先决条件**

- Healthcheck 工具仅在 RHEL 8.1 或更新的系统中可用。

## 107.1. 文件权限健康检查测试

Healthcheck 工具测试由 Identity Management(IdM)安装和配置的一些重要文件的所有权和权限。

如果您更改了任何测试文件的所有权或权限，测试会在 results 部分中 返回警告。虽然这不一定意味着配置不起作用，但这意味着 文件与默认配置不同。

要查看所有测试，请使用 --list-sources 选项运行 ipa-healthcheck：

```
# ipa-healthcheck --list-sources
```

您可以在 ipahealthcheck.ipa.files 源中找到文件权限测试：

**IPAFileNSSDBCheck**

此测试会检查 389-ds NSS 数据库和证书颁发机构(CA)数据库。389-ds 数据库位于 /etc/dirsrv/slapd-<dashed-REALM> 中，CA 数据库位于 /etc/pki/pki-tomcat/alias/ 中。

**IPAFileCheck**

此测试检查以下文件：

- /var/lib/ipa/ra-agent.{key|pem}
- /var/lib/ipa/certs/httpd.pem

- /var/lib/ipa/private/httpd.key
- /etc/httpd/alias/ipasession.key
- /etc/dirsrv/ds.keytab
- /etc/ipa/ca.crt
- /etc/ipa/custodia/server.keys

如果启用了 PKINIT：

- /var/lib/ipa/certs/kdc.pem
- /var/lib/ipa/private/kdc.key

如果配置了 DNS：

- /etc/named.keytab
- /etc/ipa/dnssec/ipa-dnskeysyncd.keytab

**TomcatFileCheck**

如果配置了 CA，则此测试会检查一些特定于 tomcat 的文件：

- /etc/pki/pki-tomcat/password.conf
- /var/lib/pki/pki-tomcat/conf/ca/CS.cfg

- /etc/pki/pki-tomcat/server.xml

注意

当尝试找到问题时，在所有 IdM 服务器中运行这些测试。

### 107.2. 使用 HEALTHCHECK 处理配置文件

按照以下流程，使用 Healthcheck 工具对身份管理(IdM)服务器配置文件运行独立的手动测试。

Healthcheck 工具包含许多测试。可以通过以下方法缩小结果：

- 排除所有成功测试 ：--failures-only
- 仅包含所有权和权限测试 ：-- source=ipahealthcheck.ipa.files

流程

1. 要在 IdM 配置文件所有权和权限中运行 Healthcheck 测试，同时只显示警告、错误和严重问题，请输入：

```
# ipa-healthcheck --source=ipahealthcheck.ipa.files --failures-only
```

成功测试会显示空括号：

```
# ipa-healthcheck --source=ipahealthcheck.ipa.files --failures-only
[]
```

失败的测试显示结果 类似如下 ：

```
{
  "source": "ipahealthcheck.ipa.files",
  "check": "IPAFileNSSDBCheck",
  "result": "WARNING",
  "kw": {
```

```
    "key": "_etc_dirsrv_slapd-EXAMPLE-TEST_pkcs11.txt_mode",
    "path": "/etc/dirsrv/slapd-EXAMPLE-TEST/pkcs11.txt",
    "type": "mode",
    "expected": "0640",
    "got": "0666",
    "msg": "Permissions of /etc/dirsrv/slapd-EXAMPLE-TEST/pkcs11.txt are 0666 and should
be 0640"
  }
}
```

**其它资源**

- 请参阅 **man ipa-healthcheck**。

# 第 108 章 使用 HEALTHCHECK 检查 IDM 复制

您可以使用 Healthcheck 工具测试身份管理(IdM)复制。

详情请查看 IdM 中的 Healthcheck。

**先决条件**

- Healthcheck 工具仅在 RHEL 8.1 或更新版本中可用。

## 108.1. 复制健康检查测试

Healthcheck 工具测试身份管理(IdM)拓扑配置，并搜索复制冲突问题。

要列出所有测试，请使用 **--list-sources** 选项运行 **ipa-healthcheck**：

```
# ipa-healthcheck --list-sources
```

拓扑测试放置在 **ipahealthcheck.ipa.topology** 和 **ipahealthcheck.ds.replication** 源 下：

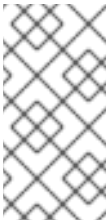
**IPATopologyDomainCheck**

此测试会验证：

- 拓扑是否没有断开连接，所有服务器之间是否存在复制路径。
- 如果服务器没有超过推荐的复制协议数。

  如果测试失败，测试会返回错误，如连接错误或太多复制协议。

  如果测试成功，则测试会返回配置的域。

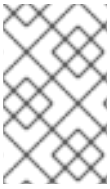


**注意**

该测试为域和 ca 后缀运行 ipa topologysuffix-verify 命令（假设在此服务器上配置了证书颁发机构）。

**ReplicationConflictCheck**

测试在 LDAP 匹配中搜索条目 （&（!(objectclass=nstombstone)）(nsds5ReplConflict=*)。



**注意**

当尝试检查问题时，在所有 IdM 服务器中运行这些测试。

有关解决 LDAP 复制冲突的更多信息，请参阅 解决常见复制问题。

## 108.2. 使用 HEALTHCHECK 进行复制

按照以下流程，使用 Healthcheck 工具对身份管理(IdM)复制拓扑和配置运行独立的手动测试。

因此，Healthcheck 工具包括了许多测试，您可以使用以下方法缩短结果：

- 复制冲突测试：-- source=ipahealthcheck.ds.replication
- 正确的拓扑测试： --source=ipahealthcheck.ipa.topology

**先决条件**

- 您必须以 root 用户身份执行 Healthcheck 测试。

**流程**

- 要运行 Healthcheck 复制冲突和拓扑检查，请输入：

```
# ipa-healthcheck --source=ipahealthcheck.ds.replication --
source=ipahealthcheck.ipa.topology
```

可能会有四种不同的结果：

- SUCCESS SAS- SAS 测试成功通过。

```
{
 "source": "ipahealthcheck.ipa.topology",
 "check": "IPATopologyDomainCheck",
 "result": "SUCCESS",
 "kw": {
  "suffix": "domain"
 }
}
```

- 预告：测试通过，但可能存在问题。

- ERROR SAS- SAS 测试失败。

```
{
 "source": "ipahealthcheck.ipa.topology",
 "check": "IPATopologyDomainCheck",
 "result": "ERROR",
 "uuid": d6ce3332-92da-423d-9818-e79f49ed321f
 "when": 20191007115449Z
 "duration": 0.005943
 "kw": {
  "msg": "topologysuffix-verify domain failed, server2 is not connected
(server2_139664377356472 in MainThread)"
 }
}
```

- CRITICAL SAS- SAS 测试失败，它会影响 IdM 服务器功能。

其它资源

- 请参阅 man ipa-healthcheck。

# 第 109 章 使用 IDM HEALTHCHECK 检查 DNS 记录

您可以使用 Healthcheck 工具识别身份管理(IdM)中的 DNS 记录的问题。

**先决条件**

- DNS 记录 Healthcheck 工具仅在 RHEL 8.2 或更新版本中可用。

## 109.1. DNS 记录健康检查测试

Healthcheck 工具包括一个测试，用于检查自动发现所需的预期 DNS 记录是否可以解析。

要列出所有测试，请使用 **--list-sources** 选项运行 **ipa-healthcheck**：

```
# ipa-healthcheck --list-sources
```

您可以在 **ipahealthcheck.ipa.idns** 源中找到 DNS 记录检查测试。

**IPADNSSystemRecordsCheck**

此测试使用 **/etc/resolv.conf** 文件中指定的第一个解析器检查 **ipa dns-update-system-records --dry-run** 命令中的 DNS 记录。记录在 IPA 服务器上测试。

## 109.2. 使用 HEALTHCHECK 工具识别 DNS 记录

按照以下流程，使用 Healthcheck 工具在身份管理(IdM)服务器中运行 DNS 记录的独立的手动测试。

Healthcheck 工具包含许多测试。通过添加 **--source ipahealthcheck.ipa.idns** 选项，可以只包含 DNS 记录测试来缩小结果范围。

**先决条件**

- 您必须以 root 用户身份执行 Healthcheck 测试。

**流程**

- 要运行 DNS 记录检查，请输入：

```
# ipa-healthcheck --source ipahealthcheck.ipa.idns
```

如果记录可以解析，测试会返回 SUCCESS，从而返回：

```
{
  "source": "ipahealthcheck.ipa.idns",
  "check": "IPADNSSystemRecordsCheck",
  "result": "SUCCESS",
  "uuid": "eb7a3b68-f6b2-4631-af01-798cac0eb018",
  "when": "20200415143339Z",
  "duration": "0.210471",
  "kw": {
    "key": "_ldap._tcp.idm.example.com.:server1.idm.example.com."
  }
}
```

例如，当记录数量与预期数目不匹配时，测试将返回 WARNING ：

```
{
  "source": "ipahealthcheck.ipa.idns",
  "check": "IPADNSSystemRecordsCheck",
  "result": "WARNING",
  "uuid": "972b7782-1616-48e0-bd5c-49a80c257895",
  "when": "20200409100614Z",
  "duration": "0.203049",
  "kw": {
    "msg": "Got {count} ipa-ca A records, expected {expected}",
    "count": 2,
    "expected": 1
  }
}
```

其它资源

- 请参阅 man ipa-healthcheck。

# 第 110 章 演示或提升隐藏副本

安装副本后，您可以配置副本是隐藏还是可见。

有关隐藏副本的详情，请参阅 隐藏副本模式。

如果副本是 CA 续订服务器，请在隐藏此副本前将服务移到另一个副本。

详情请参阅 更改和重置 IdM CA 续订服务器。

流程

- 要隐藏副本，请输入：

```
# ipa server-state replica.idm.example.com --state=hidden
```

另外，您可以使用以下命令使副本可见：

```
# ipa server-state replica.idm.example.com --state=enabled
```

要查看拓扑中所有隐藏的副本的列表，请输入：

```
# ipa config-show
```

如果启用了所有副本，命令输出不会提到隐藏的副本

# 第 111 章 IDENTITY MANAGEMENT 安全设置

了解身份管理的与安全相关功能的更多信息。

## 111.1. 身份管理如何应用默认安全设置

默认情况下，RHEL 8 中的 Identity Management(IdM)使用系统范围的加密策略。这个策略的好处是您不需要手动强化单独的 IdM 组件。

重要

红帽建议您使用系统范围的加密策略。更改单个安全设置可能会破坏 IdM 的组件。例如：RHEL 8 中的 Java 不支持 TLS 1.3 协议。因此，使用此协议可能会导致 IdM 中失败。

其它资源

- 请参阅 crypto-policies(7) 手册页。

## 111.2. IDENTITY MANAGEMENT 中的匿名 LDAP 绑定

默认情况下，启用匿名绑定到 Identity Management(IdM)LDAP 服务器。匿名绑定可以公开某些配置设置或目录值。但是，一些实用程序（如 realmd 或 较旧的 RHEL 客户端）需要启用匿名绑定来发现注册客户端时的域设置。

其它资源

- 禁用匿名绑定

## 111.3. 禁用匿名绑定

您可以使用 LDAP 工具重置 nsslapd-allow-anonymous-access 属性来禁用 Identity Management(IdM)389 Directory Server 实例上的匿名绑定。

这些是 nsslapd-allow-anonymous-access 属性的有效值：

- 在 上 ： 允许所有匿名绑定（默认）
- Rootdse ： 仅允许匿名绑定进行 DSE 信息
- off ： 不允许任何匿名绑定

红帽不推荐通过将 属性设置为 off 来完全禁止匿名绑定，因为这也会阻止外部客户端检查服务器配置。LDAP 和 Web 客户端不一定是域客户端，因此它们会匿名连接，以读取 root DSE 文件来获取连接信息。

将 nsslapd-allow-anonymous-access 属性的值更改为 rootdse，您可以允许访问 root DSE 和服务器配置而无需访问目录数据。

警告

某些客户端依赖于匿名绑定来发现 IdM 设置。另外，对于没有使用身份验证的传统客户端，compat 树可能会中断。只有在您的客户端不需要匿名绑定时才执行这个流程。

**先决条件**

- 您可以作为 Directory Manager 进行身份验证，以写入到 LDAP 服务器。
- 您可以以 root 用户身份进行身份验证以重启 IdM 服务。

**流程**

1. 将 nsslapd-allow-anonymous-access 属性更改为 rootdse。

```
$ ldapmodify -x -D "cn=Directory Manager" -W -h server.example.com -p 389
Enter LDAP Password:
dn: cn=config
changetype: modify
```

```
replace: nsslapd-allow-anonymous-access
nsslapd-allow-anonymous-access: rootdse

modifying entry "cn=config"
```

2. 重启 389 Directory 服务器实例以加载新设置。

```
# systemctl restart dirsrv.target
```

验证

- 显示 nsslapd-allow-anonymous-access 属性的值。

```
$ ldapsearch -x -D "cn=Directory Manager" -b cn=config -W -h server.example.com  -p
389 nsslapd-allow-anonymous-access | grep nsslapd-allow-anonymous-access
Enter LDAP Password:
# requesting: nsslapd-allow-anonymous-access
nsslapd-allow-anonymous-access: rootdse
```

**其它资源**

- nsslapd-allow-anonymous-access in Directory Server 11 文档
- Identity Management 中的匿名 LDAP 绑定

# 第 112 章 在 IDM 域成员中设置 SAMBA

您可以在加入到 Red Hat Identity Management (IdM)域的主机上设置 Samba。来自IdM的用户，以及来自受信任的 Active Directory(AD)域的用户(如果有的话)可以访问 Samba 提供的共享和打印机服务。

**重要**

对 IdM 域成员使用 Samba 是一种不受支持的技术预览特性，且包含了某些限制。例如，IdM 信任控制器不支持 Active Directory 全局目录服务，它们不支持使用分布式计算环境/远程过程调用(DCE/RPC)协议解析 IdM 组。因此，AD 用户只能在登录到其他 IdM 客户端时访问托管在 IdM 客户端中的 Samba 共享和打印机；登录到 Windows 机器的 AD 用户无法访问托管在 IdM 域成员中的 Samba 共享。

我们鼓励在 IdM 域成员中部署 Samba 的用户向红帽提供反馈意见。

如果 AD 域中的用户需要访问 Samba 提供的共享和打印机服务，请确保在 AD 中启用了 AES 加密类型。如需更多信息，请参阅 使用 GPO 在活动目录中启用 AES 加密类型。

**先决条件**

- 主机作为 IdM 域的客户端加入。
- IdM 服务器和客户端必须在 RHEL 8.1 或更高版本中运行。

## 112.1. 准备 IDM 域以便在域成员中安装 SAMBA

在IdM客户端上设置Samba之前，必须在IdM服务器上使用ipa-adtrust-install工具来准备IdM域。

**注意**

运行ipa-adtrust-install命令的任何系统都会自动成为 AD 信任控制器。但是，您必须在 IdM 服务器上只运行一次 ipa-adtrust-install 。

**先决条件**

- IdM 服务器已安装。
- 您需要 root 权限才能安装软件包并重新启动 IdM 服务。

**流程**

1. 安装所需的软件包：

```
[root@ipaserver ~]# yum install ipa-server-trust-ad samba-client
```

2. 以 IdM 管理用户身份进行身份验证：

```
[root@ipaserver ~]# kinit admin
```

3. 运行 ipa-adtrust-install 工具：

```
[root@ipaserver ~]# ipa-adtrust-install
```

如果 IdM 安装了集成的 DNS 服务器，则会自动创建 DNS 服务记录。

如果您在没有集成 DNS 服务器的情况下安装了 IdM，ipa-adtrust-install 会打印一个服务记录列表，您必须手动将它们添加到 DNS，然后才能继续操作。

4. 该脚本提示您 /etc/samba/smb.conf 已存在，并将被重写：

```
WARNING: The smb.conf already exists. Running ipa-adtrust-install will break your existing Samba configuration.

Do you wish to continue? [no]: yes
```

5. 该脚本提示您配置 slapi-nis 插件，这是一个兼容插件，允许旧的 Linux 客户端与受信任的用户一起工作：

```
Do you want to enable support for trusted domains in Schema Compatibility plugin? This will allow clients older than SSSD 1.9 and non-Linux clients to work with trusted users.
```

```
Enable trusted domains support in slapi-nis? [no]: yes
```

6. 提示时，输入 IdM 域的 NetBIOS 名称，或者按 Enter 接受推荐的名称：

```
Trust is configured but no NetBIOS domain name found, setting it now.
Enter the NetBIOS name for the IPA domain.
Only up to 15 uppercase ASCII letters, digits and dashes are allowed.
Example: EXAMPLE.

NetBIOS domain name [IDM]:
```

7. 系统会提示您运行 SID 生成任务，以便为任何现有用户创建 SID：

```
Do you want to run the ipa-sidgen task? [no]: yes
```

这是一个资源密集型任务，因此如果您有大量的用户，您可以在其他时间运行此操作。

8. （可选）默认情况下，对于 Windows Server 2008 及更高版本，动态 RPC 端口范围定义为 49152-65535。如果需要为您的环境定义一个不同的动态 RPC 端口范围，请将 Samba 配置为使用不同的端口，并在防火墙设置中开放这些端口。以下示例将端口范围设置为55000-65000。

```
[root@ipaserver ~]# net conf setparm global 'rpc server dynamic port range' 55000-65000
[root@ipaserver ~]# firewall-cmd --add-port=55000-65000/tcp
[root@ipaserver ~]# firewall-cmd --runtime-to-permanent
```

9. 重启ipa服务：

```
[root@ipaserver ~]# ipactl restart
```

10. 使用smbclient工具来验证 Samba 是否响应 IdM 端的 Kerberos 身份验证：

```
[root@ipaserver ~]# smbclient -L ipaserver.idm.example.com -U user_name --use-kerberos=required
lp_load_ex: changing to config backend registry
   Sharename       Type      Comment
   ---------       ----      -------
   IPC$            IPC       IPC Service (Samba 4.15.2)
...
```

### 112.2. 在 IDM 客户端中安装和配置 SAMBA 服务器

您可以在 IdM 域中注册的客户端上安装和配置 Samba。

**先决条件**

- IdM 服务器和客户端必须在 RHEL 8.1 或更高版本中运行。
- IdM 域已准备好，如 为在域成员上安装 Samba 准备 IdM 域 中所述。
- 如果 IdM 具有配置了 AD 的信任，请为 Kerberos 启用 AES 加密类型。例如，使用组策略对象(GPO)来启用 AES 加密类型。详情请参阅 使用 GPO 在活动目录中启用 AES 加密。

**流程**

1. 安装ipa-client-samba软件包：

```
[root@idm_client]# yum install ipa-client-samba
```

2. 使用ipa-client-samba工具准备客户端并创建初始 Samba 配置：

```
[root@idm_client]# ipa-client-samba
Searching for IPA server...
IPA server: DNS discovery
Chosen IPA master: idm_server.idm.example.com
SMB principal to be created: cifs/idm_client.idm.example.com@IDM.EXAMPLE.COM
NetBIOS name to be used: IDM_CLIENT
Discovered domains to use:

 Domain name: idm.example.com
NetBIOS name: IDM
        SID: S-1-5-21-525930803-952335037-206501584
   ID range: 212000000 - 212199999

 Domain name: ad.example.com
NetBIOS name: AD
        SID: None
   ID range: 1918400000 - 1918599999

Continue to configure the system with these values? [no]: yes
Samba domain member is configured. Please check configuration at
/etc/samba/smb.conf and start smb and winbind services
```

3. 默认情况下，ipa-client-samba会自动将[homes]部分添加到/etc/samba/smb.conf文件中，该文件在用户连接时动态共享用户的主目录。如果用户在这个服务器上没有主目录，或者您不想共享主目录，请从/etc/samba/smb.conf中删除以下行：

```
[homes]
   read only = no
```

4. 共享目录和打印机。详情请查看以下部分：

- 设置使用 POSIX ACL 的 Samba 文件共享
- 设置使用 Windows ACL 的共享
- 将 Samba 设置为打印服务器

5. 在本地防火墙中打开 Samba 客户端所需的端口：

```
[root@idm_client]# firewall-cmd --permanent --add-service=samba-client
[root@idm_client]# firewall-cmd --reload
```

6. 启用并启动smb和winbind服务：

```
[root@idm_client]# systemctl enable --now smb winbind
```

验证步骤

在安装了 samba-client 软件包的不同的 IdM 域成员上运行以下验证步骤：

- 使用 Kerberos 身份验证列出 Samba 服务器中的共享：

```
$ smbclient -L idm_client.idm.example.com -U user_name --use-kerberos=required
lp_load_ex: changing to config backend registry

   Sharename       Type      Comment
   ---------       ----      -------
```

```
    example         Disk
    IPC$            IPC       IPC Service (Samba 4.15.2)
...
```

其它资源

- ipa-client-samba(1) man page

### 112.3. 如果 IDM 信任新域，请手动添加 ID 映射配置

Samba 需要一个 ID 映射配置，用户可从该域访问资源。在 IdM 客户端上运行的现有 Samba 服务器上，在管理员向 Active Directory(AD)域添加了新的信任后，您必须手动添加 ID 映射配置。

先决条件

- 您在 IdM 客户端中配置了 Samba。之后，IdM 增加了一个新的信任。
- 在可信 AD 域中必须禁用 Kerberos 的 DES 和 RC4 加密类型。为了安全起见，RHEL 8 不支持这些弱加密类型。

流程

1. 使用主机的 keytab 进行身份验证：

```
[root@idm_client]# kinit -k
```

2. 使用ipa idrange-find命令来显示新域的基本 ID 和 ID 范围大小。例如，以下命令显示了ad.example.com域的值：

```
[root@idm_client]# ipa idrange-find --name="AD.EXAMPLE.COM_id_range" --raw
---------------
1 range matched
---------------
  cn: AD.EXAMPLE.COM_id_range
  ipabaseid: 1918400000
  ipaidrangesize: 200000
  ipabaserid: 0
  ipanttrusteddomainsid: S-1-5-21-968346183-862388825-1738313271
  iparangetype: ipa-ad-trust
----------------------------
Number of entries returned 1
----------------------------
```

在后续步骤中，您需要ipabaseid和ipaidrangesize属性的值。

3. 要计算可用最高的 ID，请使用以下公式：

```
maximum_range = ipabaseid + ipaidrangesize - 1
```

使用上一步中的值，ad.example.com域的最大可用 ID 是1918599999(1918400000 + 200000 - 1)。

4. 编辑/etc/samba/smb.conf文件，并将域的 ID 映射配置添加到[global]部分：

```
idmap config AD : range = 1918400000 - 1918599999
idmap config AD : backend = sss
```

将ipabaseid属性的值指定为最小值，将上一步中的计算值指定为该范围的最大值。

5. 重启smb和winbind服务：

```
[root@idm_client]# systemctl restart smb winbind
```

验证步骤

- 使用 Kerberos 身份验证列出 Samba 服务器中的共享：

```
$ smbclient -L idm_client.idm.example.com -U user_name --use-kerberos=required
lp_load_ex: changing to config backend registry

   Sharename       Type      Comment
   ---------       ----      -------
   example         Disk
   IPC$            IPC       IPC Service (Samba 4.15.2)
...
```

## 112.4. 其它资源

- 安装身份管理客户端

# 第 113 章 使用外部身份提供程序向 IDM 进行身份验证

您可以将用户与支持 OAuth 2 设备授权流的外部身份提供者(IdP)关联。当这些用户使用 RHEL 8.7 或更高版本中提供的 SSSD 版本进行身份验证时，它们会在外部 IdP 执行身份验证和授权后收到带有 Kerberos 票据的 RHEL 身份管理(IdM)单点登录的功能。

主要特性包括：

- 使用 ipa idp-* 命令添加、修改和删除对外部 IdP 的引用。
- 使用 ipa user-mod --user-auth-type=idp 命令为用户启用 IdP 身份验证。

## 113.1. 将 IDM 连接到外部 IDP 的好处

作为管理员，您可能想要允许存储在外部身份源（如云服务供应商）中的用户访问连接到 Identity Management (IdM)环境的 RHEL 系统。要达到此目的，您可以将这些用户的 Kerberos 票据的身份验证和授权过程委托给该外部实体。

您可以使用此功能扩展 IdM 的功能，并允许存储在外部身份提供程序(IdP)中的用户访问由 IdM 管理的 Linux 系统。

## 113.2. IDM 如何通过外部 IDP 融合登录

SSSD 2.7.0 包含 sssd-idp 软件包，该软件包可实施 idp Kerberos pre-authentication 方法。这个验证方法遵循 OAuth 2.0 设备授权流，将授权决策委派给外部 IdP：

1. IdM 客户端用户启动 OAuth 2.0 设备授权流，例如，通过在命令行中使用 kinit 实用程序检索 Kerberos TGT。
2. 一个特殊的代码和网站链接从授权服务器发送到 IdM KDC 后端。
3. IdM 客户端显示用户的链接和代码。在本例中，IdM 客户端会在命令行上输出链接和代码。

4. 用户在浏览器中打开网站链接，可以在另一个主机上、移动电话等：

   a. 用户输入特殊代码。

   b. 如有必要，用户登录到基于 OAuth 2.0 的 IdP。

   c. 系统将提示用户授权客户端访问信息。

5. 用户在原始设备提示符处确认访问。在这个示例中，用户在命令行中点击 Enter 键。

6. IdM KDC 后端轮询 OAuth 2.0 授权服务器以访问用户信息。

**支持什么：**

- 启用了 键盘互动 验证方法通过 SSH 远程登录，它允许调用可插拔式身份验证模块(PAM)库。
- 使用控制台通过登录服务进行本地 登录。
- 使用 kinit 实用程序检索 Kerberos ticket-granting ticket (TGT)。

**当前不支持什么：**

- 直接登录到 IdM WebUI。要登录到 IdM WebUI，您必须首先获取一个 Kerberos ticket。
- 直接登录 Cockpit WebUI。要登录 Cockpit Web UI，您必须首先获取一个 Kerberos ticket。

**其它资源**

- 对外部身份提供程序进行身份验证

- RFC 8628：OAuth 2.0 设备授权

### 113.3. 创建对外部身份提供程序的引用

要将外部身份提供程序(IdP)连接到您的身份管理(IdM)环境，请在 IdM 中创建 IdP 参考。完成此流程，根据 Keycloak 模板创建一个名为 my-keycloak-idp 的引用。如需了解更多引用模板，请参阅 IdM 中对不同外部 IdP 的引用。

**先决条件**

- 您已将 IdM 注册为外部 IdP，并获取客户端 ID。
- 您可以作为 IdM admin 帐户进行身份验证。
- 您的 IdM 服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 服务器使用 SSSD 2.7.0 或更高版本。

**流程**

1. 在 IdM 服务器中作为 IdM 管理员进行身份验证。

```
[root@server ~]# kinit admin
```

2. 根据 Keycloak 模板，创建一个名为 my-keycloak-idp 的引用，其中 --base-url 选项指定 Keycloak 服务器的 URL，格式为 server-name.$DOMAIN:$PORT/prefix。

```
[root@server ~]# ipa idp-add my-keycloak-idp \
                --provider keycloak --organization main \
                --base-url keycloak.idm.example.com:8443/auth \
                --client-id id13778
----------------------------------------------
Added Identity Provider reference "my-keycloak-idp"
----------------------------------------------
  Identity Provider reference name: my-keycloak-idp
  Authorization URI:
https://keycloak.idm.example.com:8443/auth/realms/main/protocol/openid-
connect/auth
  Device authorization URI:
```

```
https://keycloak.idm.example.com:8443/auth/realms/main/protocol/openid-connect/auth/device
  Token URI: https://keycloak.idm.example.com:8443/auth/realms/main/protocol/openid-connect/token
  User info URI: https://keycloak.idm.example.com:8443/auth/realms/main/protocol/openid-connect/userinfo
  Client identifier: ipa_oidc_client
  Scope: openid email
  External IdP user identifier attribute: email
```

验证

- 验证 ipa idp-show 命令的输出显示您创建的 IdP 引用。

```
[root@server ~]# ipa idp-show my-keycloak-idp
```

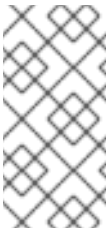
**其它资源**

- IdM 中不同外部 IdP 的引用示例
- 在 IdM 中管理外部身份提供程序的 ipa idp114 命令的选项
- ipa idp114 命令中的 --provider 选项
- ipa help idp-add

### 113.4. IDM 中不同外部 IDP 的引用示例

下表列出了 ipa idp-add 命令示例，用于在 IdM 中创建对不同 IdP 的引用。

| 身份供应商 | 重要选项 | 命令示例 |
| --- | --- | --- |
| Microsoft Identity Platform, Azure AD | **--provider microsoft --organization** | # ipa idp-add *my-azure-idp* \<br>**--provider microsoft** \<br>--organization *main* \<br>--client-id *<azure_client_id>* |

| 身份供应商 | 重要选项 | 命令示例 |
| --- | --- | --- |
| Google | **--provider google** | # ipa idp-add *my-google-idp* \<br>**--provider google** \<br>--client-id *<google_client_id>* |
| GitHub | **--provider github** | # ipa idp-add my-github-idp \<br>**--provider github** \<br>--client-id *<github_client_id>* |
| Keycloak, Red Hat Single Sign-On | **--provider keycloak**<br>**--organization**<br>**--base-url** | # ipa idp-add my-keycloak-idp \<br>**--provider keycloak** \<br>--organization main \<br>--base-url *keycloak.idm.example.com:8443/auth* \<br>--client-id *<keycloak_client_id>*<br><br>  <br>**注意**<br>Keycloak 17 及更新版本的 Quarkus 版本已删除 URI 的 **/auth/** 部分。如果您在部署中使用 Keycloak 的非Quarkus 分发，请在 **--base-url** 选项中包含 **/auth/**。 |
| Okta | **--provider okta** | # ipa idp-add my-okta-idp \<br>**--provider okta**<br>--base-url *dev-12345.okta.com* \<br>--client-id *<okta_client_id>* |

**其它资源**

- 创建对外部身份提供程序的引用
- 在 IdM 中管理外部身份提供程序的 ipa idp114 命令的选项
- ipa idp114 命令中的 --provider 选项

### 113.5. 在 IDM 中管理外部身份提供程序的 IPA IDP114 命令的选项

以下示例演示了如何根据不同的 IdP 模板配置对外部 IdP 的引用。使用以下选项指定设置：

**--provider**

其中一个已知的身份提供程序的预定义模板

**--client-id**

IdP 在应用程序注册期间发布的 OAuth 2.0 客户端标识符。当应用程序注册步骤特定于每个 IdP 时，请参考它们的文档来了解详情。如果外部 IdP 是红帽单点登录(SSO)，请参阅 创建 OpenID Connect 客户端。

**--base-url**

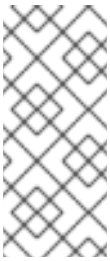
Keycloak 和 Okta 所需的 IdP 模板的基本 URL

**--organization**

Microsoft Azure 需要的 IdP 中的域或机构 ID

**--secret**

*（可选）* 如果您已将外部 IdP 配置为需要来自机密 OAuth 2.0 客户端的 secret，则使用这个选项。如果您在创建 IdP 引用时使用这个选项，则会以交互方式提示您输入 secret。将客户端 secret 作为密码保护。



**注意**

RHEL 8.7 中的 SSSD 只支持不使用客户端 secret 的非机密 OAuth 2.0 客户端。如果要使用需要机密客户端 secret 的外部 IdP，您必须在 RHEL 8.8 及之后的版本中使用 SSSD。

**其它资源**

- 创建对外部身份提供程序的引用
- IdM 中不同外部 IdP 的引用示例
- ipa idp114 命令中的 --provider 选项

### 113.6. 管理对外部 IDP 的引用

创建对外部身份提供程序(IdP)的引用后，您可以找到、显示、修改和删除该引用。本例演示了如何管理对名为 keycloak-server1 的外部 IdP 的引用。

**先决条件**

- 您可以作为 IdM admin 帐户进行身份验证。
- 您的 IdM 服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 服务器使用 SSSD 2.7.0 或更高版本。
- 您已在 IdM 中创建了对外部 IdP 的引用。请参阅创建对外部身份提供程序的引用。

**流程**

1. 在 IdM 服务器中作为 IdM 管理员进行身份验证。

```
[root@server ~]# kinit admin
```

2. 管理 IdP 参考。

   - 查找 IdP 参考，其条目包括字符串 keycloak ：

```
[root@server ~]# ipa idp-find keycloak
```

   - 显示名为 my-keycloak-idp 的 IdP 参考：

```
[root@server ~]# ipa idp-show my-keycloak-idp
```

   - 要修改 IdP 参考，请使用 ipa idp-mod 命令。例如，要更改名为 my-keycloak-idp 的 IdP 参考的 secret，请指定要提示输入 secret 的 --secret 选项：

```
[root@server ~]# ipa idp-mod my-keycloak-idp --secret
```

- 删除名为 my-keycloak-idp 的 IdP 参考：

```
[root@server ~]# ipa idp-del my-keycloak-idp
```

## 113.7. 启用 IDM 用户通过外部 IDP 进行身份验证

要启用 IdM 用户通过外部身份提供程序(IdP)，将之前创建的外部 IdP 引用与用户帐户关联。这个示例将外部 IdP 参考 keycloak-server1 与用户 idm-user-with-external-idp 关联。

**先决条件**

- 您的 IdM 客户端和服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 客户端和服务器使用 SSSD 2.7.0 或更高版本。
- 您已在 IdM 中创建了对外部 IdP 的引用。请参阅创建对外部身份提供程序的引用。

**流程**

- 修改 IdM 用户条目，将 IdP 引用与用户帐户关联：

```
[root@server ~]# ipa user-mod idm-user-with-external-idp \
            --idp my-keycloak-idp \
            --idp-user-id idm-user-with-external-idp@idm.example.com \
            --user-auth-type=idp
------------------------------
Modified user "idm-user-with-external-idp"
------------------------------
  User login: idm-user-with-external-idp
  First name: Test
  Last name: User1
  Home directory: /home/idm-user-with-external-idp
  Login shell: /bin/sh
  Principal name: idm-user-with-external-idp@idm.example.com
  Principal alias: idm-user-with-external-idp@idm.example.com
  Email address: idm-user-with-external-idp@idm.example.com
  UID: 35000003
  GID: 35000003
  User authentication types: idp
  External IdP configuration: keycloak
```

```
External IdP user identifier: idm-user-with-external-idp@idm.example.com
Account disabled: False
Password: False
Member of groups: ipausers
Kerberos keys available: False
```

验证

- 验证该用户的 ipa user-show 命令的输出是否显示对 IdP 的引用：

```
[root@server ~]# ipa user-show idm-user-with-external-idp
  User login: idm-user-with-external-idp
  First name: Test
  Last name: User1
  Home directory: /home/idm-user-with-external-idp
  Login shell: /bin/sh
  Principal name: idm-user-with-external-idp@idm.example.com
  Principal alias: idm-user-with-external-idp@idm.example.com
  Email address: idm-user-with-external-idp@idm.example.com
  ID: 35000003
  GID: 35000003
  User authentication types: idp
  External IdP configuration: keycloak
  External IdP user identifier: idm-user-with-external-idp@idm.example.com
  Account disabled: False
  Password: False
  Member of groups: ipausers
  Kerberos keys available: False
```

### 113.8. 以外部 IDP 用户身份检索 IDM TICKET-GRANTING TICKET

如果您已将身份管理(IdM)用户的身份验证委派给外部身份提供程序(IdP)，IdM 用户可以通过向外部 IdP 进行身份验证来请求 Kerberos 票据授予票据(TGT)。

完成这个流程以：

1. 在本地检索和存储匿名 Kerberos 票据。
2. 使用带有 -T 选项的 kinit 和 Secure Tunneling (FAST)频道在 idm-user-with-external-idp 用户请求 TGT，以便在 Kerberos 客户端和 Kerberos 分发中心(KDC)之间提供灵活的身份验证。

先决条件

- 您的 IdM 客户端和服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 客户端和服务器使用 SSSD 2.7.0 或更高版本。
- 您已在 IdM 中创建了对外部 IdP 的引用。请参阅创建对外部身份提供程序的引用。
- 您已与用户帐户关联了一个外部 IdP 参考。请参阅启用 IdM 用户以通过外部 IdP 进行身份验证。
- 您最初以 身份登录的用户对本地文件系统中的目录具有写入权限。

**流程**

1. 使用 Anonymous PKINIT 获取 Kerberos 票据，并将其存储在名为 ./fast.ccache 的文件中。

```
$ kinit -n -c ./fast.ccache
```

2. [可选] 查看检索到的票据：

```
$ *klist -c fast.ccache *
Ticket cache: FILE:fast.ccache
Default principal: WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS

Valid starting       Expires              Service principal
03/03/2024 13:36:37  03/04/2024 13:14:28
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
```

3. 开始以 IdM 用户身份进行身份验证，使用 -T 选项启用 FAST 通信频道。

```
[root@client ~]# kinit -T ./fast.ccache idm-user-with-external-idp
Authenticate at https://oauth2.idp.com:8443/auth/realms/master/device?
user_code=YHMQ-XKTL and press ENTER.:
```

4. 在浏览器中，以命令输出中提供的网站的用户身份进行身份验证。

5. 在命令行中，按 Enter 键来完成身份验证过程。

验证

- 显示您的 Kerberos ticket 信息，并确认行 config: pa_type shows 152 for pre-authentication with a external IdP。

```
[root@client ~]# klist -C
Ticket cache: KCM:0:58420
Default principal: idm-user-with-external-idp@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
05/09/22 07:48:23  05/10/22 07:03:07  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: fast_avail(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = yes
08/17/2022 20:22:45  08/18/2022 20:22:43
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: pa_type(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = 152
```

pa_type = 152 表示外部 IdP 身份验证。

## 113.9. 以外部 IDP 用户身份通过 SSH 登录到 IDM 客户端

要通过 SSH 作为外部身份提供程序(IdP)用户身份登录 IdM 客户端，请在命令行中开始登录过程。出现提示时，在与 IdP 关联的网站上执行身份验证过程，并在 Identity Management (IdM)客户端上完成该过程。

先决条件

- 您的 IdM 客户端和服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 客户端和服务器使用 SSSD 2.7.0 或更高版本。
- 您已在 IdM 中创建了对外部 IdP 的引用。请参阅创建对外部身份提供程序的引用。
- 您已与用户帐户关联了一个外部 IdP 参考。请参阅启用 IdM 用户以通过外部 IdP 进行身份验证。

流程

1. 尝试通过 SSH 登录到 IdM 客户端。

```
[user@client ~]$ ssh idm-user-with-external-idp@client.idm.example.com
(idm-user-with-external-idp@client.idm.example.com) Authenticate at
https://oauth2.idp.com:8443/auth/realms/main/device?user_code=XYFL-ROYR and
press ENTER.
```

2. 在浏览器中，以命令输出中提供的网站的用户身份进行身份验证。

3. 在命令行中，按 Enter 键来完成身份验证过程。

验证

- 显示您的 Kerberos ticket 信息，并确认行 config: pa_type shows 152 for pre-authentication with a external IdP。

```
[idm-user-with-external-idp@client ~]$ klist -C
Ticket cache: KCM:0:58420
Default principal: idm-user-with-external-idp@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
05/09/22 07:48:23  05/10/22 07:03:07  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: fast_avail(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = yes
08/17/2022 20:22:45  08/18/2022 20:22:43
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: pa_type(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = 152
```

## 113.10. IPA IDP114 命令中的 --PROVIDER 选项

以下身份提供程序(IdP)支持 OAuth 2.0 设备授权流：

- Microsoft Identity Platform，包括 Azure AD
- Google
- GitHub

- Keycloak，包括红帽单点登录(SSO)
- Okta

当使用 ipa idp-add 命令创建对其中一个外部 IdP 的引用时，您可以使用 --provider 选项指定 IdP 类型，它扩展至额外的选项，如下所述：

--provider=microsoft

Microsoft Azure IdP 允许基于 Azure 租户 ID 进行半虚拟化 ID，您可以使用 --organization 选项指定 ipa idp-add 命令。如果您需要对 live.com IdP 的支持，请指定 --organization common 的选项。

选择 --provider=microsoft 扩展以使用以下选项：--organization 选项的值替换了表中的字符串 ${ipaidporg}。

| 选项 | 值 |
| --- | --- |
| --auth-uri=URI | https://login.microsoftonline.com/${ipaidporg}/oauth2/v2.0/authorize |
| --dev-auth-uri=URI | https://login.microsoftonline.com/${ipaidporg}/oauth2/v2.0/devicecode |
| --token-uri=URI | https://login.microsoftonline.com/${ipaidporg}/oauth2/v2.0/token |
| --userinfo-uri=URI | https://graph.microsoft.com/oidc/userinfo |
| --keys-uri=URI | https://login.microsoftonline.com/common/discovery/v2.0/keys |
| --scope=STR | OpenID 电子邮件 |
| --idp-user-id=STR | email |

--provider=google

选择 --provider=google 扩展以使用以下选项：

| 选项 | 值 |
| --- | --- |
| --auth-uri=URI | https://accounts.google.com/o/oauth2/auth |
| --dev-auth-uri=URI | https://oauth2.googleapis.com/device/code |
| --token-uri=URI | https://oauth2.googleapis.com/token |
| --userinfo-uri=URI | https://openidconnect.googleapis.com/v1/userinfo |
| --keys-uri=URI | https://www.googleapis.com/oauth2/v3/certs |
| --scope=STR | OpenID 电子邮件 |
| --idp-user-id=STR | email |

**--provider=github**

选择 **--provider=github** 展开以使用以下选项：

| 选项 | 值 |
| --- | --- |
| --auth-uri=URI | https://github.com/login/oauth/authorize |
| --dev-auth-uri=URI | https://github.com/login/device/code |
| --token-uri=URI | https://github.com/login/oauth/access_token |
| --userinfo-uri=URI | https://openidconnect.googleapis.com/v1/userinfo |
| --keys-uri=URI | https://api.github.com/user |
| --scope=STR | user |
| --idp-user-id=STR | login |

**--provider=keycloak**

使用 Keycloak 时，您可以定义多个域或机构。由于它是自定义部署的一部分，基本 URL 和域 ID 都是必需的，因此您可以使用 **--base-url** 和 **--organization** 选项指定它们到 **ipa idp-add** 命令：

```
[root@client ~]# ipa idp-add MySSO --provider keycloak \
  --org main --base-url keycloak.domain.com:8443/auth \
  --client-id <your-client-id>
```

选择 --provider=keycloak 扩展以使用以下选项：您在 --base-url 选项中指定的值替换表中的字符串 ${ipaidpbaseurl}，而您为 --organization 'option 指定的值替换字符串 '${ipaidporg}。

| 选项 | 值 |
| --- | --- |
| --auth-uri=URI | https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/auth |
| --dev-auth-uri=URI | https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/auth/device |
| --token-uri=URI | https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/token |
| --userinfo-uri=URI | https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/userinfo |
| --scope=STR | OpenID 电子邮件 |
| --idp-user-id=STR | email |

--provider=okta

在注册一个 Okta 中的新机构后，会关联一个新的基本 URL。您可以使用 ipa idp-add 命令的 --base-url 选项指定这个基本 URL：

```
[root@client ~]# ipa idp-add MyOkta --provider okta --base-url dev-12345.okta.com --client-id <your-client-id>
```

选择 --provider=okta 扩展以使用以下选项：您为 --base-url 选项指定的值替换了表中字符串 ${ipaidpbaseurl}。

| 选项 | 值 |
| --- | --- |
| --auth-uri=URI | https://${ipaidpbaseurl}/oauth2/v1/authorize |
| --dev-auth-uri=URI | https://${ipaidpbaseurl}/oauth2/v1/device/authorize |
| --token-uri=URI | https://${ipaidpbaseurl}/oauth2/v1/token |
| --userinfo-uri=URI | https://${ipaidpbaseurl}/oauth2/v1/userinfo |
| --scope=STR | OpenID 电子邮件 |

| 选项 | 值 |
| --- | --- |
| --idp-user-id=STR | email |

其它资源

- 预填充的 IdP 模板

# 第 114 章 使用 ANSIBLE 将 IDM 用户的身份验证委派给外部身份提供程序

您可以使用 idp ansible-freeipa 模块将用户与支持 OAuth 2 设备授权流的外部身份提供程序(IdP)关联。如果存在 IdP 引用和关联的 IdP 用户 ID，您可以使用它们为用户 ansible-freeipa 模块为 IdM 用户启用 IdP 身份验证。

之后，如果这些用户使用 SSSD 版本 2.7.0 或更高版本进行身份验证，它们会在外部 IdP 执行身份验证和授权后收到带有 Kerberos 票据的 RHEL Identity Management (IdM)单点登录功能。

## 114.1. 将 IDM 连接到外部 IDP 的好处

作为管理员，您可能想要允许存储在外部身份源（如云服务供应商）中的用户访问连接到 Identity Management (IdM)环境的 RHEL 系统。要达到此目的，您可以将这些用户的 Kerberos 票据的身份验证和授权过程委托给该外部实体。

您可以使用此功能扩展 IdM 的功能，并允许存储在外部身份提供程序(IdP)中的用户访问由 IdM 管理的 Linux 系统。

## 114.2. IDM 如何通过外部 IDP 融合登录

SSSD 2.7.0 包含 sssd-idp 软件包，该软件包可实施 idp Kerberos pre-authentication 方法。这个验证方法遵循 OAuth 2.0 设备授权流，将授权决策委派给外部 IdP：

1. IdM 客户端用户启动 OAuth 2.0 设备授权流，例如，通过在命令行中使用 kinit 实用程序检索 Kerberos TGT。
2. 一个特殊的代码和网站链接从授权服务器发送到 IdM KDC 后端。
3. IdM 客户端显示用户的链接和代码。在本例中，IdM 客户端会在命令行上输出链接和代码。
4. 用户在浏览器中打开网站链接，可以在另一个主机上、移动电话等：
   a. 用户输入特殊代码。

b. 如有必要，用户登录到基于 OAuth 2.0 的 IdP。

c. 系统将提示用户授权客户端访问信息。

5. 用户在原始设备提示符处确认访问。在这个示例中，用户在命令行中点击 Enter 键。

6. IdM KDC 后端轮询 OAuth 2.0 授权服务器以访问用户信息。

支持什么：

- 启用了 键盘互动 验证方法通过 SSH 远程登录，它允许调用可插拔式身份验证模块(PAM)库。
- 使用控制台通过登录服务进行本地 登录。
- 使用 kinit 实用程序检索 Kerberos ticket-granting ticket (TGT)。

当前不支持什么：

- 直接登录到 IdM WebUI。要登录到 IdM WebUI，您必须首先获取一个 Kerberos ticket。
- 直接登录 Cockpit WebUI。要登录 Cockpit Web UI，您必须首先获取一个 Kerberos ticket。

其它资源

- 对外部身份提供程序进行身份验证
- RFC 8628：OAuth 2.0 设备授权

### 114.3. 使用 ANSIBLE 创建对外部身份提供程序的引用

要将外部身份提供程序(IdP)连接到您的身份管理(IdM)环境，请在 IdM 中创建 IdP 参考。完成此流程，

使用 idp ansible-freeipa 模块配置对 github 外部 IdP 的引用。

先决条件

- 您已将 IdM 作为 OAuth 应用程序注册到外部 IdP，并在 IdM 用户要使用的设备中生成客户端 ID 和客户端 secret，以向 IdM 进行身份验证。示例假定：
  - my_github_account_name 是 github 用户，其将 IdM 用户用于向 IdM 进行身份验证的帐户。
  - 客户端 ID 为 2efe1acffe9e8ab869f4。
  - 客户端 secret 为 656a5228abc5f9545c85fa626aecbf69312d398c。
- 您的 IdM 服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 服务器使用 SSSD 2.7.0 或更高版本。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 您使用 RHEL 8.10 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

流程

1. 在 Ansible 控制节点上，创建一个 configure-external-idp-reference.yml playbook：

```
---
- name: Configure external IdP
  hosts: ipaserver
  become: false
  gather_facts: false

  tasks:
  - name: Ensure a reference to github external provider is available
    ipaidp:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: github_idp
      provider: github
      client_ID: 2efe1acffe9e8ab869f4
      secret: 656a5228abc5f9545c85fa626aecbf69312d398c
      idp_user_id: my_github_account_name
```

2. 保存该文件。

3. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory configure-external-idp-reference.yml
```

验证

- 在 IdM 客户端上，验证 ipa idp-show 命令的输出显示您创建的 IdP 引用。

```
[idmuser@idmclient ~]$ ipa idp-show github_idp
```

后续步骤

- 使用 Ansible 启用 IdM 用户通过外部 IdP 进行身份验证

其它资源

- idp ansible-freeipa 上游文档

## 114.4. 使用 ANSIBLE 启用 IDM 用户通过外部 IDP 进行身份验证

您可以使用 用户 ansible-freeipa 模块启用身份管理(IdM)用户通过外部身份提供程序(IdP)进行身份验证。为此，请将之前创建的外部 IdP 引用与 IdM 用户帐户关联。完成此流程，以使用 Ansible 将名为 github_idp 的外部 IdP 参考与名为 idm-user-with-external-idp 的 IdM 用户关联。因此，用户可以使用 my_github_account_name github 身份作为 idm-user-with-external-idp 进行身份验证。

**先决条件**

- 您的 IdM 客户端和服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 客户端和服务器使用 SSSD 2.7.0 或更高版本。
- 您已在 IdM 中创建了对外部 IdP 的引用。请参阅使用 Ansible 创建对外部身份提供程序的引用。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 您使用 RHEL 8.10 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

**流程**

1. 在 Ansible 控制节点上，创建一个 enable-user-to-authenticate-via-external-idp.yml playbook：

```
---
- name: Ensure an IdM user uses an external IdP to authenticate to IdM
```

```
hosts: ipaserver
become: false
gather_facts: false

tasks:
- name: Retrieve Github user ID
  ansible.builtin.uri:
    url: “https://api.github.com/users/my_github_account_name”
    method: GET
    headers:
      Accept: “application/vnd.github.v3+json”
  register: user_data

- name: Ensure IdM user exists with an external IdP authentication
  ipauser:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: idm-user-with-external-idp
    first: Example
    last: User
    userauthtype: idp
    idp: github_idp
    idp_user_id: my_github_account_name
```

2. 保存该文件。

3. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory enable-user-to-authenticate-via-external-idp.yml
```

验证

- 登录到 IdM 客户端，并验证 idm-user-with-external-idp 用户的 ipa user-show 命令的输出是否显示对 IdP 的引用：

```
$ ipa user-show idm-user-with-external-idp
User login: idm-user-with-external-idp
First name: Example
Last name: User
Home directory: /home/idm-user-with-external-idp
Login shell: /bin/sh
Principal name: idm-user-with-external-idp@idm.example.com
Principal alias: idm-user-with-external-idp@idm.example.com
Email address: idm-user-with-external-idp@idm.example.com
ID: 35000003
GID: 35000003
User authentication types: idp
External IdP configuration: github
```

```
External IdP user identifier: idm-user-with-external-idp@idm.example.com
Account disabled: False
Password: False
Member of groups: ipausers
Kerberos keys available: False
```

其它资源

- idp ansible-freeipa 上游文档

### 114.5. 以外部 IDP 用户身份检索 IDM TICKET-GRANTING TICKET

如果您已将身份管理(IdM)用户的身份验证委派给外部身份提供程序(IdP)，IdM 用户可以通过向外部 IdP 进行身份验证来请求 Kerberos 票据授予票据(TGT)。

完成这个流程以：

1. 在本地检索和存储匿名 Kerberos 票据。
2. 使用带有 -T 选项的 kinit 和 Secure Tunneling (FAST)频道在 idm-user-with-external-idp 用户请求 TGT，以便在 Kerberos 客户端和 Kerberos 分发中心(KDC)之间提供灵活的身份验证。

先决条件

- 您的 IdM 客户端和服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 客户端和服务器使用 SSSD 2.7.0 或更高版本。
- 您已在 IdM 中创建了对外部 IdP 的引用。请参阅使用 Ansible 创建对外部身份提供程序的引用。
- 您已与用户帐户关联了一个外部 IdP 参考。请参阅使用 Ansible 启用 IdM 用户通过外部 IdP 进行身份验证。

- 您最初以 身份登录的用户对本地文件系统中的目录具有写入权限。

流程

1. 使用 Anonymous PKINIT 获取 Kerberos 票据，并将其存储在名为 ./fast.ccache 的文件中。

```
$ kinit -n -c ./fast.ccache
```

2. [可选] 查看检索到的票据：

```
$ *klist -c fast.ccache *
Ticket cache: FILE:fast.ccache
Default principal: WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS

Valid starting       Expires              Service principal
03/03/2024 13:36:37  03/04/2024 13:14:28
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
```

3. 开始以 IdM 用户身份进行身份验证，使用 -T 选项启用 FAST 通信频道。

```
[root@client ~]# kinit -T ./fast.ccache idm-user-with-external-idp
Authenticate at https://oauth2.idp.com:8443/auth/realms/master/device?
user_code=YHMQ-XKTL and press ENTER.:
```

4. 在浏览器中，以命令输出中提供的网站的用户身份进行身份验证。

5. 在命令行中，按 Enter 键来完成身份验证过程。

验证

- 显示您的 Kerberos ticket 信息，并确认行 config: pa_type shows 152 for pre-authentication with a external IdP。

```
[root@client ~]# klist -C
Ticket cache: KCM:0:58420
Default principal: idm-user-with-external-idp@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
05/09/22 07:48:23  05/10/22 07:03:07  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: fast_avail(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = yes
```

```
08/17/2022 20:22:45  08/18/2022 20:22:43
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: pa_type(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = 152
```

pa_type = 152 表示外部 IdP 身份验证。

### 114.6. 以外部 IDP 用户身份通过 SSH 登录到 IDM 客户端

要通过 SSH 作为外部身份提供程序(IdP)用户身份登录 IdM 客户端，请在命令行中开始登录过程。出现提示时，在与 IdP 关联的网站上执行身份验证过程，并在 Identity Management (IdM)客户端上完成该过程。

**先决条件**

- 您的 IdM 客户端和服务器使用 RHEL 8.7 或更高版本。
- 您的 IdM 客户端和服务器使用 SSSD 2.7.0 或更高版本。
- 您已在 IdM 中创建了对外部 IdP 的引用。请参阅使用 Ansible 创建对外部身份提供程序的引用。
- 您已与用户帐户关联了一个外部 IdP 参考。请参阅使用 Ansible 启用 IdM 用户通过外部 IdP 进行身份验证。

**流程**

1. 尝试通过 SSH 登录到 IdM 客户端。

```
[user@client ~]$ ssh idm-user-with-external-idp@client.idm.example.com
(idm-user-with-external-idp@client.idm.example.com) Authenticate at
https://oauth2.idp.com:8443/auth/realms/main/device?user_code=XYFL-ROYR and
press ENTER.
```

2. 在浏览器中，以命令输出中提供的网站的用户身份进行身份验证。
3. 在命令行中，按 Enter 键来完成身份验证过程。

验证

- 显示您的 Kerberos ticket 信息，并确认行 config: pa_type shows 152 for pre-authentication with a external IdP。

```
[idm-user-with-external-idp@client ~]$ klist -C
Ticket cache: KCM:0:58420
Default principal: idm-user-with-external-idp@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
05/09/22 07:48:23  05/10/22 07:03:07  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: fast_avail(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = yes
08/17/2022 20:22:45  08/18/2022 20:22:43
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: pa_type(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = 152
```

## 114.7. IPAIDP ANSIBLE 模块中的 PROVIDER 选项

以下身份提供程序(IdP)支持 OAuth 2.0 设备授权流：

- Microsoft Identity Platform，包括 Azure AD
- Google
- GitHub
- Keycloak，包括红帽单点登录(SSO)
- Okta

当使用 idp ansible-freeipa 模块创建对这些外部 IdP 的引用时，您可以使用 ipaidp ansible-freeipa playbook 任务中的 provider 选项指定 IdP 类型，它扩展至额外的选项，如下所述：

**Provider: microsoft**

Microsoft Azure IdP 允许基于 Azure 租户 ID 进行半虚拟化 ID，您可以使用 机构 选项指定。如果您需要对 live.com IdP 的支持，请指定选项 organization common。

选择 **provider: microsoft** 扩展以使用以下选项。**organization** 选项的值替换表中的字符串 **${ipaidporg}**。

| 选项 | 值 |
| --- | --- |
| **auth_uri: URI** | **https://login.microsoftonline.com/${ipaidporg}/oauth2/v2.0/authorize** |
| **dev_auth_uri: URI** | **https://login.microsoftonline.com/${ipaidporg}/oauth2/v2.0/devicecode** |
| **token_uri: URI** | **https://login.microsoftonline.com/${ipaidporg}/oauth2/v2.0/token** |
| **userinfo_uri: URI** | **https://graph.microsoft.com/oidc/userinfo** |
| **keys_uri: URI** | **https://login.microsoftonline.com/common/discovery/v2.0/keys** |
| **Scope: STR** | **OpenID 电子邮件** |
| **idp_user_id: STR** | **email** |

**Provider: google**

选择 **供应商：google** 扩展以使用以下选项：

| 选项 | 值 |
| --- | --- |
| **auth_uri: URI** | **https://accounts.google.com/o/oauth2/auth** |
| **dev_auth_uri: URI** | **https://oauth2.googleapis.com/device/code** |
| **token_uri: URI** | **https://oauth2.googleapis.com/token** |
| **userinfo_uri: URI** | **https://openidconnect.googleapis.com/v1/userinfo** |
| **keys_uri: URI** | **https://www.googleapis.com/oauth2/v3/certs** |
| **Scope: STR** | **OpenID 电子邮件** |
| **idp_user_id: STR** | **email** |

**Provider: github**

选择 **provider: github** 扩展以使用以下选项：

| 选项 | 值 |
| --- | --- |
| **auth_uri: URI** | **https://github.com/login/oauth/authorize** |
| **dev_auth_uri: URI** | **https://github.com/login/device/code** |
| **token_uri: URI** | **https://github.com/login/oauth/access_token** |
| **userinfo_uri: URI** | **https://openidconnect.googleapis.com/v1/userinfo** |
| **keys_uri: URI** | **https://api.github.com/user** |
| **Scope: STR** | **user** |
| **idp_user_id: STR** | **login** |

**provider: keycloak**

使用 Keycloak 时，您可以定义多个域或机构。由于它通常是自定义部署的一部分，因此基本 URL 和域 ID 都是必需的，因此您可以使用 **ipaidp** playbook 任务中的 **base_url** 和 **机构** 选项指定它们：

```
---
- name: Playbook to manage IPA idp
  hosts: ipaserver
  become: false

  tasks:
  - name: Ensure keycloak idp my-keycloak-idp is present using provider
    ipaidp:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: my-keycloak-idp
      provider: keycloak
      organization: main
      base_url: keycloak.domain.com:8443/auth
      client_id: my-keycloak-client-id
```

选择 **provider: keycloak** 扩展以使用以下选项。您在 **base_url** 选项中指定的值替换表中的字符串 **${ipaidpbaseurl}**，您为 **机构"option** 指定的值替换字符串 **'${ipaidporg}**。

| 选项 | 值 |
| --- | --- |
| **auth_uri: URI** | **https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/auth** |

| 选项 | 值 |
| --- | --- |
| dev_auth_uri: URI | https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/auth/device |
| token_uri: URI | https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/token |
| userinfo_uri: URI | https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/userinfo |
| Scope: STR | OpenID 电子邮件 |
| idp_user_id: STR | email |

**Provider: okta**

在注册一个 Okta 中的新机构后，会关联一个新的基本 URL。您可以使用 ipaidp playbook 任务中的 base_url 选项指定这个基本 URL：

```
---
- name: Playbook to manage IPA idp
  hosts: ipaserver
  become: false

  tasks:
  - name: Ensure okta idp my-okta-idp is present using provider
    ipaidp:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: my-okta-idp
      provider: okta
      base_url: dev-12345.okta.com
      client_id: my-okta-client-id
```

选择 provider: okta 扩展以使用以下选项。为 base_url 选项指定的值替换表中的字符串 ${ipaidpbaseurl}。

| 选项 | 值 |
| --- | --- |
| auth_uri: URI | https://${ipaidpbaseurl}/oauth2/v1/authorize |
| dev_auth_uri: URI | https://${ipaidpbaseurl}/oauth2/v1/device/authorize |

| 选项 | 值 |
| --- | --- |
| **token_uri: URI** | **https://${ipaidpbaseurl}/oauth2/v1/token** |
| **userinfo_uri: URI** | **https://${ipaidpbaseurl}/oauth2/v1/userinfo** |
| **Scope: STR** | **OpenID 电子邮件** |
| **idp_user_id: STR** | **email** |

**其它资源**

- **预填充的 IdP 模板**

# 第 115 章 IDM 与其他红帽产品的集成

以下链接是与 IdM 集成的其他红帽产品的文档。您可以配置这些产品，以允许 IdM 用户可以访问它们的服务。

**Ansible Automation Platform**

设置 LDAP 身份验证

**OpenShift Container Platform**

配置 LDAP 身份提供程序

**OpenStack Platform**

将 OpenStack 身份(keystone)与红帽身份管理器(IdM)集成

**Satellite**

使用红帽身份管理

**单点登录**

SSSD 和 FreeIPA 身份管理集成

**虚拟化**

配置外部 LDAP 提供商

# 第 116 章 使用 ANSIBLE 将 IDM 与 NIS 域和网络组集成

## 116.1. NIS 及其优点

在 UNIX 环境中，网络信息服务(NIS)是一种集中管理身份和身份验证的通用方法。NIS 最初被命名为 Yellow Pages (YP)，其集中管理身份验证和身份信息，例如：

- 用户和密码
- 主机名和 IP 地址
- POSIX 组

对于现代网络基础架构，NIS 被视为不安全的，例如，它既不提供主机身份验证，也不会通过网络发送加密数据。要临时解决这个问题，NIS 通常与其他协议集成以增强安全性。

如果您使用身份管理(IdM)，您可以使用 NIS 服务器插件连接无法完全迁移到 IdM 的客户端。IdM 将网络组和其他 NIS 数据集成到 IdM 域中。另外，您可以轻松地将用户和主机身份从 NIS 域迁移到 IdM。

netgroups 可在 NIS 组期望的任何位置使用。

其它资源

- IdM 中的 NIS
- IdM 中的 NIS netgroups
- 从 NIS 迁移到身份管理

## 116.2. IDM 中的 NIS

IdM 中的 NIS 对象

NIS 对象集成并存储在目录服务器后端中，符合 RFC 2307。IdM 在 LDAP 目录中创建 NIS 对象，客户端使用加密的 LDAP 连接，通过例如：系统安全服务守护进程(SSSD)或 nss_ldap 检索它们。

IdM 管理网络组、帐户、组、主机和其他数据。IdM 使用 NIS 侦听器将密码、组和网络组映射到 IdM 条目。

### IdM 中的 NIS 插件

对于 NIS 支持，IdM 使用 slapi-nis 软件包提供的以下插件：

#### NIS 服务器插件

NIS 服务器插件使 IdM 集成的 LDAP 服务器充当客户端的 NIS 服务器。在此角色中，目录服务器会根据配置动态生成和更新 NIS 映射。使用插件，IdM 将使用 NIS 协议的客户端用作 NIS 服务器。

#### 模式兼容性插件

模式兼容性插件可让目录服务器后端提供一个存储在目录信息树(DIT)部分中的条目的替代视图。这包括添加、丢弃或重命名属性值，以及选择性地从树中的多个条目检索属性值。

详情请查看 /usr/share/doc/slapi-nis-*version*/sch-getting-started.txt 文件。

## 116.3. IDM 中的 NIS NETGROUPS

NIS 实体可以存储在网络组中。与 UNIX 组相比，网络组为以下内容提供支持：

- 嵌套组（作为其他组成员的组）。
- 分组主机。

netgroup 定义一组以下信息： host、user 和 domain。这个集合被称为 triple。这三个字段可以包含：

- 值。

- 短划线(-)，指定 "no valid value"
- 无值。空字段指定一个通配符。

```
(host.example.com,,nisdomain.example.com)
(-,user,nisdomain.example.com)
```

当客户端请求 NIS netgroup 时，IdM 会将 LDAP 条目转换：

- 到传统的 NIS 映射，并使用 NIS 插件将其发送到客户端。
- 与 RFC 2307 或 RFC 2307bis 兼容的 LDAP 格式。

## 116.4. 使用 ANSIBLE 确保 NETGROUP 存在

您可以使用 Ansible playbook 确保 IdM netgroup 存在。这个示例描述了如何确包 TestNetgroup1 组存在。

### 先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。

### 流程

1. 使用以下内容创建您的 Ansible playbook 文件 netgroup-present.yml ：

```
---
- name: Playbook to manage IPA netgroup.
  hosts: ipaserver
  become: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure netgroup members are present
    ipanetgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: TestNetgroup1
```

2. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/netgroup-
present.yml
```

**其它资源**

- IdM 中的 NIS
- /usr/share/doc/ansible-freeipa/README-netgroup.md
- /usr/share/doc/ansible-freeipa/playbooks/netgroup

### 116.5. 使用 ANSIBLE 确保 NETGROUP 中存在成员

您可以使用 Ansible playbook 确保 IdM 用户、组和网络组是 netgroup 的成员。这个示例描述了如何确保 TestNetgroup1 组具有以下成员：

- user1 和 user2 IdM 用户
- group1 IdM 组

- admins netgroup
- 是 IdM 客户端的 idmclient1 主机

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。
- TestNetgroup1 IdM netgroup 存在。
- user1 和 user2 IdM 用户已存在。
- group1 IdM 组已存在。
- admins IdM netgroup 存在。

**流程**

1. 使用以下内容创建 Ansible playbook 文件 IdM-members-present-in-a-netgroup.yml ：

```
---
- name: Playbook to manage IPA netgroup.
  hosts: ipaserver
  become: no
```

```
vars_files:
- /home/user_name/MyPlaybooks/secret.yml
tasks:
- name: Ensure netgroup members are present
  ipanetgroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: TestNetgroup1
    user: user1,user2
    group: group1
    host: idmclient1
    netgroup: admins
    action: member
```

2. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/ldM-
members-present-in-a-netgroup.yml
```

其它资源

- IdM 中的 NIS
- /usr/share/doc/ansible-freeipa/README-netgroup.md
- /usr/share/doc/ansible-freeipa/playbooks/netgroup

### 116.6. 使用 ANSIBLE 确保一个成员从 NETGROUP 中删除

您可以使用 Ansible playbook 确保 IdM 用户是 netgroup 的成员。这个示例描述了如何确保 TestNetgroup1 组在其 members. netgroup 中没有 user1 IdM 用户。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。

- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
- 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
- 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。

- TestNetgroup1 netgroup 存在。

**流程**

1. 使用以下内容创建 Ansible playbook 文件 IdM-member-absent-from-a-netgroup.yml ：

```
---
- name: Playbook to manage IPA netgroup.
  hosts: ipaserver
  become: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure netgroup user, "user1", is absent
    ipanetgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: TestNetgroup1
      user: "user1"
      action: member
      state: absent
```

2. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/IdM-
member-absent-from-a-netgroup.yml
```

**其它资源**

- IdM 中的 NIS

- /usr/share/doc/ansible-freeipa/README-netgroup.md
- /usr/share/doc/ansible-freeipa/playbooks/netgroup

### 116.7. 使用 ANSIBLE 确保没有 NETGROUP

您可以使用 Ansible playbook 确保身份管理(IdM)中不存在 netgroup。这个示例描述了如何确保 IdM 域中不存在 TestNetgroup1 组。

**先决条件**

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 您已在 ~/*MyPlaybooks*/ 目录中创建了一个具有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。

**流程**

1. 使用以下内容创建您的 Ansible playbook 文件 netgroup-absent.yml ：

```
---
- name: Playbook to manage IPA netgroup.
  hosts: ipaserver
  become: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure netgroup my_netgroup1 is absent
    ipanetgroup:
```

```
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: my_netgroup1
    state: absent
```

2. 运行 **playbook:**

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/netgroup-
absent.yml
```

**其它资源**

- IdM 中的 NIS
- /usr/share/doc/ansible-freeipa/README-netgroup.md
- /usr/share/doc/ansible-freeipa/playbooks/netgroup

# 第 117 章 从 NIS 迁移到身份管理

网络信息服务(NIS)服务器可以包含有关用户、组、主机、网络组和自动挂载映射的信息。作为系统管理员，您可以将这些条目类型、身份验证和授权从 NIS 服务器迁移到身份管理(IdM)服务器，以便在 IdM 服务器上执行所有用户管理操作。从 NIS 迁移到 IdM 还允许您访问更为安全的协议，如 Kerberos。

## 117.1. 在 IDM 中启用 NIS

要允许 NIS 和 Identity Management(IdM)服务器之间的通信，您必须在 IdM 服务器中启用 NIS 兼容性选项。

**先决条件**

- 在 IdM 服务器中具有 root 访问权限。

**流程**

1. 在 IdM 服务器中启用 NIS 侦听程序和兼容性插件：

```
[root@ipaserver ~]# ipa-nis-manage enable
[root@ipaserver ~]# ipa-compat-manage enable
```

2. *可选* ：对于更严格的防火墙配置，请设置固定的端口。

例如，将端口设置为未使用的端口 514 ：

```
[root@ipaserver ~]# ldapmodify -x -D 'cn=directory manager' -W
dn: cn=NIS Server,cn=plugins,cn=config
changetype: modify
add: nsslapd-pluginarg0
nsslapd-pluginarg0: 514
```

**警告**

为避免与其他服务冲突，请勿使用任何 1024 以上的端口号。

3. 启用并启动端口映射器服务：

```
[root@ipaserver ~]# systemctl enable rpcbind.service
[root@ipaserver ~]# systemctl start rpcbind.service
```

4. 重启目录服务器：

```
[root@ipaserver ~]# systemctl restart dirsrv.target
```

### 117.2. 将用户条目从 NIS 迁移到 IDM

NIS passwd 映射包含有关用户的信息，如名称、UID、主组、GECOS、shell 和主目录。使用此数据将 NIS 用户帐户迁移到身份管理(IdM)：

**先决条件**

- 在 NIS 服务器中具有 root 访问权限.
- 在 IdM 中启用了 NIS。
- NIS 服务器已加入 IdM。

**流程**

1. 安装 yp-tools 软件包：

```
[root@nis-server ~]# yum install yp-tools -y
```

2. 在 NIS 服务器中 创建包含以下内容的 /root/nis-users.sh 脚本：

```
#!/bin/sh
# $1 is the NIS domain, $2 is the primary NIS server
ypcat -d $1 -h $2 passwd > /dev/shm/nis-map.passwd 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.passwd) ; do
 IFS=' '
 username=$(echo $line | cut -f1 -d:)
```

```
# Not collecting encrypted password because we need cleartext password
# to create kerberos key
uid=$(echo $line | cut -f3 -d:)
gid=$(echo $line | cut -f4 -d:)
gecos=$(echo $line | cut -f5 -d:)
homedir=$(echo $line | cut -f6 -d:)
shell=$(echo $line | cut -f7 -d:)

# Now create this entry
echo passw0rd1 | ipa user-add $username --first=NIS --last=USER \
    --password --gidnumber=$gid --uid=$uid --gecos="$gecos" --homedir=$homedir \
    --shell=$shell
ipa user-show $username
done
```

3. 以 IdM admin 用户身份进行身份验证：

```
[root@nis-server ~]# kinit admin
```

4. 运行脚本。例如：

```
[root@nis-server ~]# sh /root/nis-users.sh nisdomain nis-server.example.com
```

**重要**

此脚本对名字和姓氏使用硬编码的值，并将密码设置为 passw0rd1。用户在下次登录时必须更改临时密码。

## 117.3. 将用户组从 NIS 迁移到 IDM

NIS 组映射包含有关组的信息，如组名称、GID 或组成员。使用此数据将 NIS 组迁移到身份管理 (IdM)：

**先决条件**

- 在 NIS 服务器中具有 root 访问权限.
- 在 IdM 中启用了 NIS。
- NIS 服务器已加入 IdM。

**流程**

1. 安装 yp-tools 软件包：

```
[root@nis-server ~]# yum install yp-tools -y
```

2. 在 NIS 服务器中使用以下内容创建 /root/nis-groups.sh 脚本：

```
#!/bin/sh
# $1 is the NIS domain, $2 is the primary NIS server
ypcat -d $1 -h $2 group > /dev/shm/nis-map.group 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.group); do
 IFS=' '
 groupname=$(echo $line | cut -f1 -d:)
 # Not collecting encrypted password because we need cleartext password
 # to create kerberos key
 gid=$(echo $line | cut -f3 -d:)
 members=$(echo $line | cut -f4 -d:)

 # Now create this entry
 ipa group-add $groupname --desc=NIS_GROUP_$groupname --gid=$gid
 if [ -n "$members" ]; then
  ipa group-add-member $groupname --users={$members}
 fi
 ipa group-show $groupname
done
```

3. 以 IdM admin 用户身份进行身份验证：

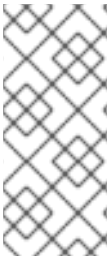
```
[root@nis-server ~]# kinit admin
```

4. 运行脚本。例如：

```
[root@nis-server ~]# sh /root/nis-groups.sh nisdomain nis-server.example.com
```

## 117.4. 将主机条目从 NIS 迁移到 IDM

NIS 主机 映射包含有关主机的信息，如主机名和 IP 地址。使用此数据将 NIS 主机条目迁移到身份管理 (IdM)：



**注意**

当您在 IdM 中创建主机组时，会自动创建对应的 shadow NIS 组。不要在这些影子 NIS 组中使用 ipa netgroup-* 命令。使用 ipa netgroup-* 命令仅管理通过 netgroup-add 命令创建的原生网络组。

**先决条件**

- 在 NIS 服务器中具有 root 访问权限.
- 在 IdM 中启用了 NIS。
- NIS 服务器已加入 IdM。

**流程**

1. 安装 yp-tools 软件包：

```
[root@nis-server ~]# yum install yp-tools -y
```

2. 在 NIS 服务器中使用以下内容创建 /root/nis-hosts.sh 脚本：

```
#!/bin/sh
# $1 is the NIS domain, $2 is the primary NIS server
ypcat -d $1 -h $2 hosts | egrep -v "localhost|127.0.0.1" > /dev/shm/nis-map.hosts 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.hosts); do
 IFS=' '
 ipaddress=$(echo $line | awk '{print $1}')
 hostname=$(echo $line | awk '{print $2}')
 primary=$(ipa env xmlrpc_uri | tr -d '[:space:]' | cut -f3 -d: | cut -f3 -d/)
 domain=$(ipa env domain | tr -d '[:space:]' | cut -f2 -d:)
 if [ $(echo $hostname | grep "\." |wc -l) -eq 0 ] ; then
  hostname=$(echo $hostname.$domain)
 fi
 zone=$(echo $hostname | cut -f2- -d.)
 if [ $(ipa dnszone-show $zone 2>/dev/null | wc -l) -eq 0 ] ; then
  ipa dnszone-add --name-server=$primary --admin-email=root.$primary
 fi
 ptrzone=$(echo $ipaddress | awk -F. '{print $3 "." $2 "." $1 ".in-addr.arpa."}')
 if [ $(ipa dnszone-show $ptrzone 2>/dev/null | wc -l) -eq 0 ] ; then
  ipa dnszone-add  $ptrzone --name-server=$primary --admin-email=root.$primary
```

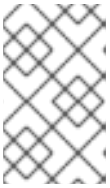
```
fi
# Now create this entry
ipa host-add $hostname --ip-address=$ipaddress
ipa host-show $hostname
done
```

3. 以 IdM admin 用户身份进行身份验证：

```
[root@nis-server ~]# kinit admin
```

4. 运行脚本。例如：

```
[root@nis-server ~]# sh /root/nis-hosts.sh nisdomain nis-server.example.com
```



**注意**

此脚本不会迁移特殊的主机配置，如别名。

## 117.5. 将 NETGROUP 条目从 NIS 迁移到 IDM

NIS netgroup 映射包含有关网络组的信息。使用此数据将 NIS 网络组迁移到身份管理(IdM)：

**先决条件**

- 在 NIS 服务器中具有 root 访问权限.
- 在 IdM 中启用了 NIS。
- NIS 服务器已加入 IdM。

**流程**

1. 安装 yp-tools 软件包：

```
[root@nis-server ~]# yum install yp-tools -y
```

2. 在 NIS 服务器中使用以下内容创建 /root/nis-netgroups.sh 脚本：

```
#!/bin/sh
# $1 is the NIS domain, $2 is the primary NIS server
ypcat -k -d $1 -h $2 netgroup > /dev/shm/nis-map.netgroup 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.netgroup); do
 IFS=' '
 netgroupname=$(echo $line | awk '{print $1}')
 triples=$(echo $line | sed "s/^$netgroupname //")
 echo "ipa netgroup-add $netgroupname --desc=NIS_NG_$netgroupname"
 if [ $(echo $line | grep "(," | wc -l) -gt 0 ]; then
  echo "ipa netgroup-mod $netgroupname --hostcat=all"
 fi
 if [ $(echo $line | grep ",," | wc -l) -gt 0 ]; then
  echo "ipa netgroup-mod $netgroupname --usercat=all"
 fi

 for triple in $triples; do
  triple=$(echo $triple | sed -e 's/-//g' -e 's/(//' -e 's/)//')
  if [ $(echo $triple | grep ",.*," | wc -l) -gt 0 ]; then
   hostname=$(echo $triple | cut -f1 -d,)
   username=$(echo $triple | cut -f2 -d,)
   domain=$(echo $triple | cut -f3 -d,)
   hosts=""; users=""; doms="";
   [ -n "$hostname" ] && hosts="--hosts=$hostname"
   [ -n "$username" ] && users="--users=$username"
   [ -n "$domain"   ] && doms="--nisdomain=$domain"
   echo "ipa netgroup-add-member $netgroup $hosts $users $doms"
  else
   netgroup=$triple
   echo "ipa netgroup-add $netgroup --desc=<NIS_NG>_$netgroup"
  fi
 done
done
```

3. 以 IdM admin 用户身份进行身份验证：

```
[root@nis-server ~]# kinit admin
```

4. 运行脚本。例如：

```
[root@nis-server ~]# sh /root/nis-netgroups.sh nisdomain nis-server.example.com
```

## 117.6. 将自动挂载映射从 NIS 迁移到 IDM

自动挂载映射是一系列嵌套的条目，它们定义位置（父条目）、关联的键和映射。将 NIS 自动挂载映射到身份管理(IdM)：

### 先决条件

- 在 NIS 服务器中具有 root 访问权限.
- 在 IdM 中启用了 NIS。
- NIS 服务器已加入 IdM。

### 流程

1. 安装 yp-tools 软件包：

```
[root@nis-server ~]# yum install yp-tools -y
```

2. 使用 NIS 服务器的以下内容创建 /root/nis-automounts.sh 脚本：

```
#!/bin/sh
# $1 is for the automount entry in ipa

ipa automountlocation-add $1

# $2 is the NIS domain, $3 is the primary NIS server, $4 is the map name

ypcat -k -d $2 -h $3 $4 > /dev/shm/nis-map.$4 2>&1

ipa automountmap-add $1 $4

basedn=$(ipa env basedn | tr -d '[:space:]' | cut -f2 -d:)
cat > /tmp/amap.ldif <<EOF
dn: nis-domain=$2+nis-map=$4,cn=NIS Server,cn=plugins,cn=config
objectClass: extensibleObject
nis-domain: $2
nis-map: $4
nis-base: automountmapname=$4,cn=$1,cn=automount,$basedn
nis-filter: (objectclass=\*)
nis-key-format: %{automountKey}
nis-value-format: %{automountInformation}
EOF
ldapadd -x -h $3 -D "cn=Directory Manager" -W -f /tmp/amap.ldif

IFS=$'\n'
```

```
for line in $(cat /dev/shm/nis-map.$4); do
 IFS=" "
 key=$(echo "$line" | awk '{print $1}')
 info=$(echo "$line" | sed -e "s^$key[ \t]*"")
 ipa automountkey-add nis $4 --key="$key" --info="$info"
done
```

**注意**

脚本导出 NIS 自动挂载信息，为自动挂载位置和相关映射生成 LDAP 数据交换格式(LDIF)，并将 LDIF 文件导入到 IdM 目录服务器。

3. 以 IdM admin 用户身份进行身份验证：

```
[root@nis-server ~]# kinit admin
```

4. 运行脚本。例如：

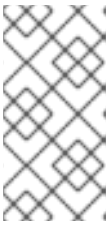
```
[root@nis-server ~]# sh /root/nis-automounts.sh location nisdomain
    nis-server.example.com map_name
```

# 第 118 章 在 IDM 中使用自动挂载

自动挂载是在多个系统间管理、组织和访问目录的一种方式。每当请求访问一个目录时，Automount 会自动挂载该目录。这在身份管理(IdM)域中工作良好，因为它允许您在域中的客户端上轻松共享目录。

这个示例使用以下场景：

- nfs-server.idm.example.com 是网络文件系统(NFS)服务器的完全限定域名(FQDN)。
- 为了简单起见，nfs-server.idm.example.com 是一个 IdM 客户端，为 raleigh 自动挂载位置提供映射。



**注意**

自动挂载位置是一组唯一的 NFS 映射。理想情况下，这些映射都位于同一地理位置，例如：客户端可以从快速连接中受益，但这不是强制要求。

- NFS 服务器以读写形式导出 /exports/project 目录。
- 属于 developers 组的任何 IdM 用户都可以在使用 raleigh 自动挂载位置的任何 IdM 客户端中以 /devel/project/ 来访问导出的目录。
- IdM -client.idm.example.com 是一个 IdM 客户端，它使用 raleigh 自动挂载位置。

**重要**

如果要使用 Samba 服务器而不是 NFS 服务器来为 IdM 客户端提供共享，请参阅 如何在 IPA 环境中使用 Autofs 配置进行过Kerberos 的 CIFS 挂载？KCS 解决方案。

## 118.1. IDM 中的 AUTOFS 和自动挂载

autofs 服务可根据需要自动化目录的挂载，方法是在目录被访问时，将 automount 守护进程定向到挂载目录。此外，在不活动一段时间后，autofs 将automount 定向到未卸载的自动挂载的目录。与静态挂载不同，按需挂载可节省系统资源。

自动挂载映射

在使用 autofs 的系统上，automount 配置存储在几个不同的文件中。主要的 automount 配置文件是 /etc/auto.master，其中包含系统上 automount 的主映射以及相关的资源。此映射称为 *自动挂载映射*。

/etc/auto.master 配置文件包含 *主映射*。它可以包含对其他映射的引用。这些映射可以是直接的，也可以是间接的。直接映射使用挂载点的绝对路径名，而间接映射则使用相对路径名。

IdM 中的自动挂载配置

虽然 automount 通常从本地 /etc/auto.master 和相关文件检索其映射数据，但它也可以从其他源检索映射数据。一个通用源是 LDAP 服务器。在身份管理(IdM)环境中，这是一个 389 目录服务器。

如果使用 autofs 的系统是 IdM 域中的一个客户端，则 automount 配置不会存储在本地配置文件中。相反，autofs 配置（如映射、位置和密钥）作为 LDAP 条目存储在 IdM 目录中。例如，对于 idm.example.com IdM 域，默认的 *主映射* 存储如下：

```
dn:
automountmapname=auto.master,cn=default,cn=automount,dc=idm,dc=example,dc=com
objectClass: automountMap
objectClass: top
automountMapName: auto.master
```

其它资源

- 根据需要挂载文件系统

## 118.2. 在 RED HAT IDENTITY MANAGEMENT 域中使用 KERBEROS 建立一个 NFS 服务器

如果您使用 Red Hat Identity Management (IdM)，您可以将 NFS 服务器加入到 IdM 域中。这可让您集中管理用户和组，并使用 Kerberos 进行身份验证、完整性保护和流量加密。

先决条件

- NFS 服务器在 Red Hat Identity Management (IdM)域中 已注册。
- NFS 服务器正在运行并已配置。

流程

1. 以 IdM 管理员身份获取 kerberos 票据：

```
# kinit admin
```

2. 创建一个 nfs/<FQDN> 服务主体：

```
# ipa service-add nfs/nfs_server.idm.example.com
```

3. 从 IdM 检索 nfs 服务主体，并将其存储在 /etc/krb5.keytab 文件中：

```
# ipa-getkeytab -s idm_server.idm.example.com -p nfs/nfs_server.idm.example.com -k
/etc/krb5.keytab
```

4. 可选：显示 /etc/krb5.keytab 文件中的主体：

```
# klist -k /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
```

默认情况下，当您将主机加入到 IdM 域时，IdM 客户端会将主机主体添加到 /etc/krb5.keytab 文件中。如果缺少主机主体，请使用 ipa-getkeytab -s *idm_server.idm.example.com* -p *host/nfs_server.idm.example.com* -k /etc/krb5.keytab 命令添加它。

5. 使用 ipa-client-automount 工具配置 IdM ID 的映射：

```
#  ipa-client-automount
Searching for IPA server...
IPA server: DNS discovery
Location: default
Continue to configure the system with these values? [no]: yes
```

```
Configured /etc/idmapd.conf
Restarting sssd, waiting for it to become available.
Started autofs
```

6. 更新 /etc/exports 文件，并将 Kerberos 安全方法添加到客户端选项中。例如：

```
/nfs/projects/       192.0.2.0/24(rw,sec=krb5i)
```

如果您希望客户端可以从多个安全方法中选择，请使用冒号分割它们：

```
/nfs/projects/       192.0.2.0/24(rw,sec=krb5:krb5i:krb5p)
```

7. 重新载入导出的文件系统：

```
# exportfs -r
```

### 118.3. 使用 IDM CLI 在 IDM 中配置自动挂载位置和映射

位置是一组映射，全部存储在 auto.master 中。一个位置可以存储多个映射。位置条目仅充当映射条目的容器；它本身并不是一个自动挂载配置。

作为身份管理(IdM)中的系统管理员，您可以在 IdM 中配置自动挂载位置和映射，以便指定位置中的 IdM 用户可以通过导航到其主机上的特定挂载点来访问 NFS 服务器导出的共享。导出的 NFS 服务器目录和挂载点都在映射中指定。这个示例描述了如何配置 raleigh 位置和映射，其将 nfs-server.idm.example.com:/exports/project 共享作为读写目录，挂载到 IdM 客户端上的 /devel/ 挂载点。

**先决条件**

- 您以 IdM 管理员的身份登录到任何注册了 IdM 的主机上。

**流程**

1. 创建 raleigh 自动挂载位置：

```
$ ipa automountlocation-add raleigh
----------------------------------
Added automount location "raleigh"
```

```
-------------------------------
 Location: raleigh
```

2. 在 raleigh 位置创建一个 auto.devel 自动挂载映射：

```
$ ipa automountmap-add raleigh auto.devel
------------------------------
Added automount map "auto.devel"
------------------------------
 Map: auto.devel
```

3. 添加 exports/ 共享的密钥和挂载信息：

   a. 为 auto.devel 映射添加密钥和挂载信息：

```
$ ipa automountkey-add raleigh auto.devel --key='*' --info='-sec=krb5p,vers=4 nfs-server.idm.example.com:/exports/&'
---------------------
Added automount key "*"
---------------------
 Key: *
 Mount information: -sec=krb5p,vers=4 nfs-server.idm.example.com:/exports/&
```

   b. 为 auto.master 映射添加密钥和挂载信息：

```
$ ipa automountkey-add raleigh auto.master --key=/devel --info=auto.devel
--------------------------
Added automount key "/devel"
--------------------------
 Key: /devel
 Mount information: auto.devel
```

## 118.4. 在 IDM 客户端上配置自动挂载

作为身份管理(IdM)系统管理员，您可以在 IdM 客户端上配置自动挂载服务，以便在用户登录客户端时 IdM 用户可以自动访问为已添加客户端的位置配置的 NFS 共享。这个示例描述了如何配置 IdM 客户端，以使用 raleigh 位置中可用的 automount 服务。

**先决条件**

- 您有访问 IdM 客户端的 root 权限。

- 以 IdM 管理员身份登录。
- 自动挂载位置存在。示例位置为 raleigh。

**流程**

1. 在 IdM 客户端上，输入 ipa-client-automount 命令并指定位置。使用 -U 选项以无人值守方式运行脚本：

```
# ipa-client-automount --location raleigh -U
```

2. 停止 autofs 服务，清除 SSSD 缓存，然后启动 autofs 服务来加载新的配置设置：

```
# systemctl stop autofs ; sss_cache -E ; systemctl start autofs
```

## 118.5. 验证 IDM 用户能否访问 IDM 客户端上的 NFS 共享

作为身份管理(IdM)系统管理员，您可以在登录到特定的 IdM 客户端时测试作为特定组一员的 IdM 用户是否可以访问 NFS 共享。

在示例中，测试了以下场景：

- 属于 developers 组的名为 idm_user 的 IdM 用户可以读写自动挂载在 idm-client.idm.example.com （一个位于 raleigh 自动挂载位置的 IdM 客户端）上的/devel/project 目录中的内容。

**先决条件**

- 您已 在 IdM 主机上建立了一个带有 Kerberos 的 NFS 服务器。
- 您已 在 IdM 中配置了自动挂载位置、映射和挂载点，您在其中配置了 IdM 用户如何访问 NFS 共享。
- 您已 在 IdM 客户端上配置了自动挂载。

**流程**

1. 验证 IdM 用户能否可以访问 读-写 目录：

   a. 以 IdM 用户身份连接到 IdM 客户端：

   ```
   $ ssh idm_user@idm-client.idm.example.com
   Password:
   ```

   b. 获取 IdM 用户的票据授权票据(TGT)：

   ```
   $ kinit idm_user
   ```

   c. [可选] 查看 IdM 用户的组成员身份：

   ```
   $ ipa user-show idm_user
     User login: idm_user
     [...]
     Member of groups: developers, ipausers
   ```

   d. 进入到 /devel/project 目录：

   ```
   $ cd /devel/project
   ```

   e. 列出目录内容：

   ```
   $ ls
   rw_file
   ```

   f. 对目录中的文件添加一行来测试 写 权限：

   ```
   $ echo "idm_user can write into the file" > rw_file
   ```

   g. [可选] 查看更新的文件内容：

   ```
   $ cat rw_file
   this is a read-write file
   idm_user can write into the file
   ```

输出确认 idm_user 可以对该文件进行写入。

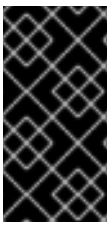
# 第 119 章 使用 ANSIBLE 为 IDM 用户自动挂载 NFS 共享

自动挂载是在多个系统间管理、组织和访问目录的一种方式。每当请求访问一个目录时，Automount 会自动挂载该目录。这在身份管理(IdM)域中工作良好，因为它允许您在域中的客户端上轻松共享目录。

您可以使用 Ansible 配置 NFS 共享，以使其可以被 IdM 位置中登录到 IdM 客户端的 IdM 用户自动挂载。

本章中的示例使用以下场景：

- nfs-server.idm.example.com 是网络文件系统(NFS)服务器的完全限定域名(FQDN)。
- nfs-server.idm.example.com 是位于 raleigh 自动挂载位置的 IdM 客户端。
- NFS 服务器以读写形式导出 /exports/project 目录。
- 属于 developers 组的任何 IdM 用户都可以访问导出的目录的内容，因为 IdM 客户端上的 /devel/project/ 位于与 NFS 服务器相同的 raleigh 自动挂载位置。
- idm-client.idm.example.com 是位于 raleigh 自动挂载位置的 IdM 客户端。



重要

如果要使用 Samba 服务器而不是 NFS 服务器来为 IdM 客户端提供共享，请参阅 如何在 IPA 环境中使用 Autofs 配置进行过Kerberos 的 CIFS 挂载？KCS 解决方案。

本章包含以下部分：

1. IdM 中的 autofs 和自动挂载

2. 在 IdM 中建立一个具有 Kerberos 的 NFS 服务器

3. 使用 Ansible 在 IdM 中配置自动挂载位置、映射和密钥

4. 使用 Ansible 将 IdM 用户添加到拥有 NFS 共享的组中

5. 在 IdM 客户端上配置自动挂载

6. 验证 IdM 用户能否访问 IdM 客户端上的 NFS 共享

## 119.1. IDM 中的 AUTOFS 和自动挂载

autofs 服务可根据需要自动化目录的挂载，方法是在目录被访问时，将 automount 守护进程定向到挂载目录。此外，在不活动一段时间后，autofs 将automount 定向到未卸载的自动挂载的目录。与静态挂载不同，按需挂载可节省系统资源。

### 自动挂载映射

在使用 autofs 的系统上，automount 配置存储在几个不同的文件中。主要的 automount 配置文件是 /etc/auto.master，其中包含系统上 automount 的主映射以及相关的资源。此映射称为 *自动挂载映射*。

/etc/auto.master 配置文件包含 *主映射*。它可以包含对其他映射的引用。这些映射可以是直接的，也可以是间接的。直接映射使用挂载点的绝对路径名，而间接映射则使用相对路径名。

### IdM 中的自动挂载配置

虽然 automount 通常从本地 /etc/auto.master 和相关文件检索其映射数据，但它也可以从其他源检索映射数据。一个通用源是 LDAP 服务器。在身份管理(IdM)环境中，这是一个 389 目录服务器。

如果使用 autofs 的系统是 IdM 域中的一个客户端，则 automount 配置不会存储在本地配置文件中。相反，autofs 配置（如映射、位置和密钥）作为 LDAP 条目存储在 IdM 目录中。例如，对于 idm.example.com IdM 域，默认的 *主映射* 存储如下：

```
dn:
automountmapname=auto.master,cn=default,cn=automount,dc=idm,dc=example,dc=com
objectClass: automountMap
```

```
objectClass: top
automountMapName: auto.master
```

**其它资源**

- 根据需要挂载文件系统

## 119.2. 在 RED HAT IDENTITY MANAGEMENT 域中使用 KERBEROS 建立一个 NFS 服务器

如果您使用 Red Hat Identity Management (IdM)，您可以将 NFS 服务器加入到 IdM 域中。这可让您集中管理用户和组，并使用 Kerberos 进行身份验证、完整性保护和流量加密。

**先决条件**

- NFS 服务器在 Red Hat Identity Management (IdM)域中 已注册。
- NFS 服务器正在运行并已配置。

**流程**

1. 以 IdM 管理员身份获取 kerberos 票据：

```
# kinit admin
```

2. 创建一个 nfs/<FQDN> 服务主体：

```
# ipa service-add nfs/nfs_server.idm.example.com
```

3. 从 IdM 检索 nfs 服务主体，并将其存储在 /etc/krb5.keytab 文件中：

```
# ipa-getkeytab -s idm_server.idm.example.com -p nfs/nfs_server.idm.example.com -k
/etc/krb5.keytab
```

4. 可选：显示 /etc/krb5.keytab 文件中的主体：

```
# klist -k /etc/krb5.keytab
```

```
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
```

默认情况下，当您将主机加入到 IdM 域时，IdM 客户端会将主机主体添加到 /etc/krb5.keytab 文件中。如果缺少主机主体，请使用 ipa-getkeytab -s *idm_server.idm.example.com* -p *host/nfs_server.idm.example.com* -k /etc/krb5.keytab 命令添加它。

5. 使用 ipa-client-automount 工具配置 IdM ID 的映射：

```
#  ipa-client-automount
Searching for IPA server...
IPA server: DNS discovery
Location: default
Continue to configure the system with these values? [no]: yes
Configured /etc/idmapd.conf
Restarting sssd, waiting for it to become available.
Started autofs
```

6. 更新 /etc/exports 文件，并将 Kerberos 安全方法添加到客户端选项中。例如：

```
/nfs/projects/       192.0.2.0/24(rw,sec=krb5i)
```

如果您希望客户端可以从多个安全方法中选择，请使用冒号分割它们：

```
/nfs/projects/       192.0.2.0/24(rw,sec=krb5:krb5i:krb5p)
```

7. 重新载入导出的文件系统：

```
# exportfs -r
```

## 119.3. 使用 ANSIBLE 在 IDM 中配置自动挂载位置、映射和密钥

作为身份管理(IdM)系统管理员，您可以在 IdM 中配置自动挂载位置和映射，以便指定位置中的 IdM 用

户可以通过导航到其主机上的特定挂载点来访问 NFS 服务器导出的共享。导出的 NFS 服务器目录和挂载点都在映射中指定。在 LDAP 术语中，位置是此类映射条目的一个容器。

这个示例描述了如何使用 Ansible 来配置 raleigh 位置和映射，其将 nfs-server.idm.example.com:/exports/project 共享作为读写目录挂载到 IdM 客户端上的 /devel/project 挂载点。

**先决条件**

- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 在 Ansible 控制节点上，导航到 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 复制位于 /usr/share/doc/ansible-freeipa/playbooks/automount/ 目录中的 automount-location-present.yml Ansible playbook 文件：

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automount/automount-location-present.yml automount-location-map-and-key-present.yml
```

3. 打开 automount-location-map-and-key-present.yml 文件进行编辑。

4. 通过在 ipaautomountlocation 任务部分设置以下变量来调整文件：

   - 将 ipaadmin_password 变量设为 IdM admin 的密码。
   - 将 name 变量设为 raleigh。
   - 确保 state 变量设置为 present。

   这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Automount location present example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure automount location is present
    ipaautomountlocation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: raleigh
      state: present
```

5. 继续编辑 automount-location-map-and-key-present.yml 文件：

   a. 在 tasks 部分中，添加一个任务来确保存在一个自动挂载映射：

```
[...]
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
[...]
  - name: ensure map named auto.devel in location raleigh is created
    ipaautomountmap:
      ipaadmin_password: "{{ ipaadmin_password }}"
```

```
    name: auto.devel
    location: raleigh
    state: present
```

b. 添加另一个任务，将挂载点和 NFS 服务器信息添加到映射：

```
[...]
 vars_files:
 - /home/user_name/MyPlaybooks/secret.yml
 tasks:
[...]
 - name: ensure automount key /devel/project is present
   ipaautomountkey:
     ipaadmin_password: "{{ ipaadmin_password }}"
     location: raleigh
     mapname: auto.devel
     key: /devel/project
     info: nfs-server.idm.example.com:/exports/project
     state: present
```

c. 添加另一个任务以确保 auto.devel 已连接到 auto.master ：

```
[...]
 vars_files:
 - /home/user_name/MyPlaybooks/secret.yml
 tasks:
[...]
- name: Ensure auto.devel is connected in auto.master:
  ipaautomountkey:
    ipaadmin_password: "{{ ipaadmin_password }}"
    location: raleigh
    mapname: auto.map
    key: /devel
    info: auto.devel
    state: present
```

6. 保存该文件。

7. 运行 Ansible playbook ， 并指定 playbook 和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automount-
location-map-and-key-present.yml
```

## 119.4. 使用 ANSIBLE 将 IDM 用户添加到拥有 NFS 共享的组中

作为身份管理(IdM)系统管理员，您可以使用 Ansible 来创建可以访问 NFS 共享的用户组，并将 IdM 用户添加到此组中。

本例描述了如何使用 Ansible playbook 来确保 idm_user 帐户属于 developers 组，以便 idm_user 可以访问 /exports/project NFS 共享。

**先决条件**

- 您有访问 nfs-server.idm.example.com NFS 服务器的 root 权限，该服务器是一个位于 raleigh 自动挂载位置的 IdM 客户端。
- 您需要知道 IdM admin 密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。
  - 在 ~/*MyPlaybooks*/ 中，您已创建了automount-location-map-and-key-present.yml 文件，该文件已包含 使用 Ansible 在 IdM 中配置自动挂载位置、映射和密钥 中的任务。

**流程**

1. 在 Ansible 控制节点上，进到 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 打开 automount-location-map-and-key-present.yml 文件进行编辑。

3. 在 tasks 部分，添加一个任务来确保 IdM developers 组存在，并且 idm_user 已添加到此组中：

```
[...]
 vars_files:
 - /home/user_name/MyPlaybooks/secret.yml
 tasks:
[...]
 - ipagroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
     name: developers
     user:
     - idm_user
     state: present
```

4. 保存该文件。

5. 运行 Ansible playbook，并指定 playbook 和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automount-location-map-and-key-present.yml
```

6. 在 NFS 服务器上，将 /exports/project 目录的组所有权更改为 developers，以便组中的每个 IdM 用户都可以访问该目录：

```
# chgrp developers /exports/project
```

## 119.5. 在 IDM 客户端上配置自动挂载

作为身份管理(IdM)系统管理员，您可以在 IdM 客户端上配置自动挂载服务，以便在用户登录客户端时 IdM 用户可以自动访问为已添加客户端的位置配置的 NFS 共享。这个示例描述了如何配置 IdM 客户端，以使用 raleigh 位置中可用的 automount 服务。

**先决条件**

- 您有访问 IdM 客户端的 root 权限。

- 以 IdM 管理员身份登录。
- 自动挂载位置存在。示例位置为 raleigh。

**流程**

1. 在 IdM 客户端上，输入 ipa-client-automount 命令并指定位置。使用 -U 选项以无人值守方式运行脚本：

```
# ipa-client-automount --location raleigh -U
```

2. 停止 autofs 服务，清除 SSSD 缓存，然后启动 autofs 服务来加载新的配置设置：

```
# systemctl stop autofs ; sss_cache -E ; systemctl start autofs
```

## 119.6. 验证 IDM 用户能否访问 IDM 客户端上的 NFS 共享

作为身份管理(IdM)系统管理员，您可以在登录到特定的 IdM 客户端时测试作为特定组一员的 IdM 用户是否可以访问 NFS 共享。

在示例中，测试了以下场景：

- 属于 developers 组的名为 idm_user 的 IdM 用户可以读写自动挂载在 idm-client.idm.example.com （一个位于 raleigh 自动挂载位置的 IdM 客户端）上的/devel/project 目录中的内容。

**先决条件**

- 您已 在 IdM 主机上建立了一个具有 Kerberos 的 NFS 服务器。
- 您已 在 IdM 中配置了自动挂载位置、映射和挂载点，您已在其中配置了 IdM 用户如何访问 NFS 共享。
- 您已 使用 Ansible 将 IdM 用户添加到拥有 NFS 共享的 developers 组中。

- 您已 在 IdM 客户端上配置了自动挂载。

**流程**

1. 验证 IdM 用户能否可以访问 读-写 目录：

   a. 以 IdM 用户身份连接到 IdM 客户端：

   ```
   $ ssh idm_user@idm-client.idm.example.com
   Password:
   ```

   b. 获取 IdM 用户的票据授权票据(TGT)：

   ```
   $ kinit idm_user
   ```

   c. [可选] 查看 IdM 用户的组成员身份：

   ```
   $ ipa user-show idm_user
     User login: idm_user
     [...]
     Member of groups: developers, ipausers
   ```

   d. 进入到 /devel/project 目录：

   ```
   $ cd /devel/project
   ```

   e. 列出目录内容：

   ```
   $ ls
   rw_file
   ```

   f. 对目录中的文件添加一行来测试 写 权限：

   ```
   $ echo "idm_user can write into the file" > rw_file
   ```

   g. [可选] 查看更新的文件内容：

```
$ cat rw_file
this is a read-write file
idm_user can write into the file
```

输出确认 idm_user 可以对该文件进行写入。

# 第 120 章 IDM 日志文件和目录

使用以下小节来监控、分析并排除 Identity Management(IdM)的独立组件：

- LDAP
- Apache Web 服务器
- 证书系统
- Kerberos
- DNS
- custodia

另外，您可以监控、分析 IdM 服务器和客户端， 并在 IdM 服务器 中启用审计日志。

## 120.1. IDM 服务器和客户端日志文件和目录

下表显示 Identity Management(IdM)服务器和客户端用来记录信息的目录和文件。您可以使用文件和目录排除安装错误。

| 目录或文件 | 描述 |
| --- | --- |
| /var/log/ipaserver-install.log | IdM 服务器的安装日志。 |
| /var/log/ipareplica-install.log | IdM 副本的安装日志。 |
| /var/log/ipaclient-install.log | IdM 客户端的安装日志。 |
| /var/log/sssd/ | SSSD 的日志文件。您可以 在 sssd.conf 文件中为 SSSD 启用详细日志记录 或 使用 ssctl 命令。 |

| 目录或文件 | 描述 |
| --- | --- |
| **~/.ipa/log/cli.log** | 日志文件，用于远程过程调用(RPC)返回的错误以及 **ipa** 实用程序的响应。在主目录中为运行工具 **的有效用**户 创建。此用户可能具有与 IdM 用户主体不同的用户名，这是在试图执行失败的 **ipa** 命令前获取 ticket(TGT)的 IdM 用户。例如，如果您以 **root** 身份登录系统，并且获取了 IdM **admin** 的 TGT，则错误会登录到 **/root/.ipa/log/cli.log** 文件。 |
| **/etc/logrotate.d/** | DNS、SSSD、Apache、Tomcat 和 Kerberos 的日志轮转策略。 |
| **/etc/pki/pki-tomcat/logging.properties** | 这个链接指向 **/usr/share/pki/server/conf/logging.properties** 的默认证书颁发机构日志记录配置。 |

**其它资源**

- IdM 服务器安装故障排除
- IdM 客户端安装故障排除
- IdM 副本安装故障排除
- IdM 中 SSSD 身份验证故障排除

### 120.2. 目录服务器日志文件

下表显示 **Identity Management(IdM)**目录服务器**(DS)**实例用来记录信息的目录和文件。您可以使用文件和目录对 **DS** 相关问题进行故障排除。

**表 120.1. 目录服务器日志文件**

| 目录或文件 | 描述 |
| --- | --- |
| **/var/log/dirsrv/slapd-*REALM_NAME*/** | 与 IdM 服务器使用的 DS 实例关联的日志文件。这里记录的大多数操作数据都与服务器数据交互相关。 |

| 目录或文件 | 描述 |
| --- | --- |
| /var/log/dirsrv/slapd-*REALM_NAME*/audit | 包含在 DS 配置中启用审计时所有 DS 操作的审计跟踪。<br><br>**注意**<br><br>您还可以审核 IdM API 日志访问中的 Apache 错误日志。但是，因为更改也可以直接使用 LDAP，因此红帽建议为审计目的启用更全面的 **/var/log/dirsrv/slapd-REALM_NAME/audit** 日志。 |
| /var/log/dirsrv/slapd-*REALM_NAME*/access | 包含有关域 DS 实例试图访问的详细信息。 |
| /var/log/dirsrv/slapd-*REALM_NAME*/errors | 包含有关域 DS 实例的失败操作的详细信息。 |

**其它资源**

- 监控服务器和数据库活动
- 日志文件参考

## 120.3. 在 IDM 服务器中启用审计日志记录

**按照以下流程，为审计目的在身份管理(IdM)服务器上启用日志记录。使用详细的日志，您可以监控数据、对问题进行故障排除，以及检查网络上的可疑活动。**

**注意**

**如果记录了大量 LDAP 更改，则 LDAP 服务可能会变得较慢，特别是在值较大时。**

**先决条件**

- **Directory Manager 密码**

**流程**

1. 绑定到 LDAP 服务器：

```
$ ldapmodify -D "cn=Directory Manager" -W << EOF
```

2. 按 [Enter]。

3. 指定您要进行的所有修改，例如：

```
dn: cn=config
changetype: modify
replace: nsslapd-auditlog-logging-enabled
nsslapd-auditlog-logging-enabled: on
-
replace:nsslapd-auditlog
nsslapd-auditlog: /var/log/dirsrv/slapd-REALM_NAME/audit
-
replace:nsslapd-auditlog-mode
nsslapd-auditlog-mode: 600
-
replace:nsslapd-auditlog-maxlogsize
nsslapd-auditlog-maxlogsize: 100
-
replace:nsslapd-auditlog-logrotationtime
nsslapd-auditlog-logrotationtime: 1
-
replace:nsslapd-auditlog-logrotationtimeunit
nsslapd-auditlog-logrotationtimeunit: day
```

4. 通过在新行中输入 EOF 来指示 ldapmodify 命令的末尾。

5. 按 [Enter] 两次。

6. 在您要在其上启用审计日志的所有其他 IdM 服务器中重复前面的步骤。

验证

- 打开 /var/log/dirsrv/slapd-REALM_NAME/audit 文件：

```
389-Directory/1.4.3.231 B2021.322.1803
server.idm.example.com:636 (/etc/dirsrv/slapd-IDM-EXAMPLE-COM)
```

```
time: 20220607102705
dn: cn=config
result: 0
changetype: modify
replace: nsslapd-auditlog-logging-enabled
nsslapd-auditlog-logging-enabled: on
[...]
```

该文件不再为空，确认启用了审计。

重要

系统会记录一个更改的条目的绑定 LDAP 可分辨名称(DN)。因此，您可能必须在对日志进行后处理。例如，在 IdM Directory 服务器中，它是一个 ID 覆盖 DN，它代表修改记录的 AD 用户的身份：

```
$ modifiersName: ipaanchoruuid=:sid:s-1-5-21-19610888-1443184010-
1631745340-279100,cn=default trust
view,cn=views,cn=accounts,dc=idma,dc=idm,dc=example,dc=com
```

如果您有用户 SID，请使用 pysss_nss_idmap.getnamebysid Python 命令查找 AD 用户：

```
>>> import pysss_nss_idmap
>>> pysss_nss_idmap.getnamebysid('S-1-5-21-1273159419-3736181166-
4190138427-500'))
{'S-1-5-21-1273159419-3736181166-4190138427-500': {'name':
'administrator@ad.vm', 'type': 3}}
```

其它资源

- Red Hat Directory Server 文档中的 Core 服务器配置属性中的 审计日志配置选项
- 如何在 IPA/IDM 服务器和 Replica Servers KCS 解决方案中启用审计日志记录
- 目录服务器日志文件

## 120.4. 修改 IDM 服务器中的错误日志

按照以下流程获取有关特定类型的错误的调试信息。该示例重点是通过将错误日志级别设置为 8192 来获得有关复制的详细错误日志。要记录不同类型的信息，请在 Red Hat Directory Server 文档中的 Error

Log Logging Levels 中选择与表不同的数字。

注意

如果记录很多 LDAP 错误，则 LDAP 服务可能会变得慢，特别是在值较大时。

先决条件

- Directory Manager 密码。

流程

1. 绑定到 LDAP 服务器：

```
$ ldapmodify -x -D "cn=directory manager" -w <password>
```

2. 按 [Enter]。

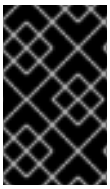
3. 指定要进行的修改。例如，仅收集与复制相关的日志：

```
dn: cn=config
changetype: modify
add: nsslapd-errorlog-level
nsslapd-errorlog-level: 8192
```

4. 按 [Enter] 两次表示 ldapmodify 指令的末尾。这将显示 修改条目 "cn=config" 消息。

5. 按 [Ctrl+C] 退出 ldapmodify 命令。

6. 在您要在其上收集关于复制错误的详细日志的其他 IdM 服务器中重复前面的步骤。



重要

完成故障排除后，将 nsslapd-errorlog-level 设置为 0 以防止性能问题。

**其它资源**

- Directory 服务器错误日志记录级别

### 120.5. IDM APACHE 服务器日志文件

下表显示 Identity Management(IdM)Apache 服务器用来记录信息的目录和文件。

**表 120.2. Apache 服务器日志文件**

| 目录或文件 | 描述 |
| --- | --- |
| **/var/log/httpd/** | Apache Web 服务器的日志文件。 |
| **/var/log/httpd/access_log** | Apache 服务器的标准访问和错误日志。特定于 IdM 的消息会和 Apache 信息一起记录，因为 IdM Web UI 和 RPC 命令行界面使用 Apache。访问日志主要仅用于用户主体和使用的 URI，通常是 RPC 端点。错误日志包含 IdM 服务器日志。 |
| **/var/log/httpd/error_log** | |

**其它资源**

- Apache 文档中的日志文件

### 120.6. IDM 中的证书系统日志文件

下表显示 Identity Management(IdM)证书系统用来记录信息的目录和文件。

**表 120.3. 证书系统日志文件**

| 目录或文件 | 描述 |
| --- | --- |
| **/var/log/pki/pki-ca-spawn.*time_of_installation*.log** | IdM 证书颁发机构(CA)的安装日志。 |
| **/var/log/pki/pki-kra-spawn.*time_of_installation*.log** | IdM 密钥恢复授权(KRA)的安装日志。 |
| **/var/log/pki/pki-tomcat/** | PKI 操作日志的顶级目录。包含 CA 和 KRA 日志。 |
| **/var/log/pki/pki-tomcat/ca/** | 包含与证书操作相关的日志的目录。在 IdM 中，这些日志用于服务主体、主机以及使用证书的其他实体。 |

| 目录或文件 | 描述 |
| --- | --- |
| **/var/log/pki/pki-tomcat/kra** | 包含与 KRA 相关的日志的目录。 |
| **/var/log/messages** | 包括其它系统消息中的证书错误消息。 |

**其它资源**

- 在 Red Hat Certificate System *Administration Guide* 中配置子系统日志

### 120.7. IDM 中的 KERBEROS 日志文件

**下表列出了 Kerberos 用来在 Identity Management(IdM)中记录信息的目录和文件。**

**表 120.4. Kerberos 日志文件**

| 目录或文件 | 描述 |
| --- | --- |
| **/var/log/krb5kdc.log** | Kerberos KDC 服务器的主日志文件。 |
| **/var/log/kadmind.log** | Kerberos 管理服务器的主日志文件。 |
| 这些文件的位置在 **krb5.conf** 文件中配置。在某些系统中，它们可能会有所不同。 | |

### 120.8. IDM 中的 DNS 日志文件

**下表列出了 DNS 用来在 Identity Management(IdM)中记录信息的目录和文件。**

**表 120.5. DNS 日志文件**

| 目录或文件 | 描述 |
| --- | --- |
| **/var/log/messages** | 包括 DNS 错误消息和其他系统信息。默认情况下不启用此文件中的 DNS 日志记录。要启用它，请输入 **# /usr/sbin/rndc querylog** 命令。该命令生成添加到 **var/log/messages** 中的以下行：<br><br>**Jun 26 17:37:33 r8server named-pkcs11[1445]: received control channel command 'querylog'**<br><br>**Jun26 17:37:33 r8server named-pkcs11[1445]: query logging now on**<br><br>要禁用日志记录，请再次运行命令。 |

### 120.9. IDM 中的 CUSTODIA 日志文件

下表显示了 Custodia 用来记录 Identity Management(IdM)中的目录和文件。

表 120.6. custodia 日志文件

| 目录或文件 | 描述 |
| --- | --- |
| /var/log/custodia/ | Custodia 服务的日志文件目录。 |

### 120.10. 其它资源

- 查看日志文件.您可以使用 journalctl 查看 systemd 单元文件的日志输出。

# 第 121 章 为 IDM 域中的 RHEL 8 WEB 控制台配置单点登录

了解如何使用 RHEL 8 web 控制台中的 Identity Management(IdM)提供的单点登录(SSO)身份验证。

优点：

- IdM 域管理员可以使用 RHEL 8 web 控制台来管理本地机器。
- IdM 域中具有 Kerberos 票据的用户不需要提供登录凭证来访问 Web 控制台。
- IdM 域已知的所有主机均可通过 RHEL 8 web 控制台本地实例的 SSH 访问。
- 不需要证书配置。控制台的 Web 服务器会自动切换到 IdM 证书颁发机构发布的证书，并被浏览器接受。

本章论述了配置用于登录到 RHEL web 控制台的 SSO 的步骤：

1. 使用 RHEL 8 web 控制台将机器添加到 IdM 域中。

   详情请参阅使用 Web 控制台将 RHEL 8 系统添加到 IdM 域中。

2. 如果要使用 Kerberos 进行身份验证，则需要在机器上获得 Kerberos ticket。

   详情请参阅使用 Kerberos 身份验证登录到 web 控制台。

3. 允许 IdM 服务器上的管理员在任何主机上运行任何命令。

   详情请参阅为 IdM 服务器上的域管理员启用管理员的 admin sudo 访问权限

先决条件

- 在 RHEL 8 系统上安装的 RHEL web 控制台。

  详情请参阅安装 Web 控制台。

- 在使用 RHEL web 控制台的系统中安装 IdM 客户端。

  详情请查看 IdM 客户端安装。

## 121.1. 使用 WEB 控制台将 RHEL 8 系统添加到 IDM 域中

您可以使用 Web 控制台将 Red Hat Enterprise Linux 8 系统添加到 Identity Management(IdM)域中。

### 先决条件

- IdM 域正在运行，并可访问您想要加入的客户端。
- 您有 IdM 域管理员凭证。

### 流程

1. 登录到 RHEL web 控制台。

   详情请参阅 Web 控制台的日志记录。

2. 在 Overview 选项卡的 Configuration 字段中点 Join Domain。

3. 在 Join a Domain 对话框的 Domain Address 字段中输入 IdM 服务器的主机名。

4. 在 Domain administrator name 字段中输入 IdM 管理帐户的用户名。

5. 在域 Domain administrator password 中添加密码。

6. 点 Join。

验证步骤

1. 如果 RHEL 8 web 控制台没有显示错误，该系统就被加入到 IdM 域，您可以在 系统 屏幕中看到域名。

2. 要验证该用户是否为域的成员，点 Terminal 页面并输入 id 命令：

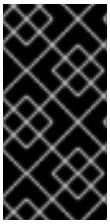
```
$ id
euid=548800004(example_user) gid=548800004(example_user)
groups=548800004(example_user)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

其它资源

- 规划身份管理
- 安装身份管理

### 121.2. 使用 KERBEROS 身份验证登录到 WEB 控制台

以下流程描述了如何设置 RHEL 8 系统以使用 Kerberos 验证的步骤。



重要

使用 SSO 时，通常在 Web 控制台中拥有任何管理特权。这只有在您配置了免密码 sudo 时有效。Web 控制台不以交互方式询问 sudo 密码。

先决条件

- IdM 域在您的公司环境中运行并可访问。

详情请参阅使用 Web 控制台将 RHEL 8 系统添加到 IdM 域中。

- 在您要通过 RHEL web 控制台连接和管理的远程系统中启用 cockpit.socket 服务。

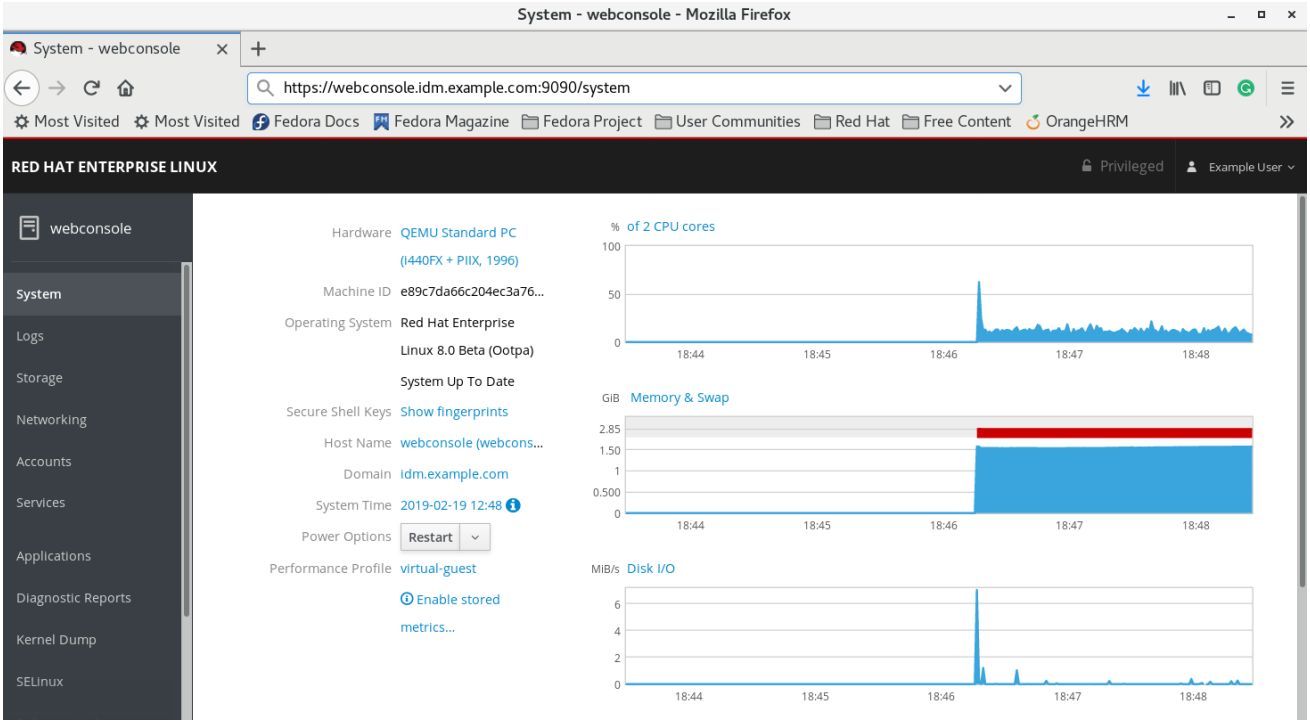
  详情请参阅安装 Web 控制台。

- 如果系统没有使用 SSSD 客户端管理的 Kerberos ticket，请尝试使用 kinit 程序手动请求 ticket。

**流程**

使用以下地址登录到 RHEL web 控制台：https://dns_name:9090

此时，您已成功连接到 RHEL web 控制台，您可以使用配置启动。




## 121.3. 为 IDM 服务器上的域管理员启用管理员 SUDO 访问权限

您可以使用 RHEL web 控制台，允许域管理员在身份管理(IdM)域中的任何主机上使用任何命令。

要实现这一目的，请启用对 IdM 服务器安装过程中自动创建的 admins 用户组的 sudo 访问权限。如果您在组上运行 ipa-advise 脚本，则添加到 admins 组的所有用户都会获得 sudo 访问权限。

**先决条件**

- 服务器运行 IdM 4.7.1 或更高版本。

**流程**

1. 连接到 IdM 服务器。

2. 运行 ipa-advise 脚本：

```
$ ipa-advise enable-admins-sudo | sh -ex
```

如果控制台没有显示错误，则 admins 组对 IdM 域中的所有机器都有 sudo 权限。

# 第 122 章 在 IDM 中使用受限委托

了解更多有关如何在身份管理(IdM)中使用受限委托功能的信息：

- 身份管理中的受限委派 描述了受限委派的工作方式。
- 配置 Web 控制台以允许使用智能卡进行身份验证的用户 SSH 到远程主机，而无需再次进行身份验证 描述了使用 Red Hat Enterprise Linux web 控制台 SSH 到远程主机，而无需进行身份验证情况下的受限委托的用例。
- 使用 Ansible 配置 Web 控制台，以允许通过智能卡进行身份验证的用户可以 SSH 到远程主机，而无需再次要求进行身份验证 描述了使用 Ansible 配置使用 Red Hat Enterprise Linux Web 控制台 SSH 到远程主机，而无需进行身份验证情况下的受限委托的用例。
- 配置 Web 控制台客户端以允许通过智能卡进行身份验证来运行 sudo，而无需身份验证即可进行身份验证， 描述了使用 Red Hat Enterprise Linux Web 控制台使用 Red Hat Enterprise Linux web 控制台运行 sudo 的情况，以便在不需要身份验证的情况下运行 sudo。
- 使用 Ansible 配置 Web 控制台，以使通过智能卡进行身份验证的用户能够运行 sudo ，而无需再次要求进行身份验证 描述了使用 Ansible 配置 Red Hat Enterprise Linux web 控制台以运行 sudo ，而无需进行身份验证情况下的受限委托的用例。

## 122.1. 身份管理中的受限委托

User for User to Proxy (S4U2proxy)扩展提供了一个服务，它代表用户为另一个服务获得服务票据。此功能称为 受限委托。第二个服务通常是代表用户授权上下文下第一个服务执行某种工作的代理。使用受限委托用户无需委托其完整的 ticket-granting ticket (TGT)。

身份管理(IdM)通常使用 Kerberos S4U2proxy 功能来允许 Web 服务器框架为用户获取 LDAP 服务票据。IdM-AD 信任系统也使用受限委托来获取 cifs 主体。

您可以使用 S4U2proxy 功能配置 web 控制台客户端，允许使用智能卡进行身份验证的 IdM 用户实现：

- 在 RHEL 主机上运行具有超级用户权限的命令，其中运行 web 控制台服务而无需再次进行身份验证。

- 使用 SSH 访问远程主机并访问主机上的服务，而无需再次进行身份验证。

其它资源

- 使用 Ansible 配置 Web 控制台，允许用户使用智能卡通过 SSH 向远程主机进行身份验证，而无需再次进行身份验证
- 使用 Ansible 配置 Web 控制台，允许用户使用智能卡进行身份验证的用户运行 sudo，而无需再次进行身份验证
- S4U2proxy
- 服务受限委派

## 122.2. 配置 WEB 控制台以允许通过智能卡通过 SSH 向远程主机进行身份验证的用户，而无需再次进行身份验证

在 RHEL web 控制台中登录到用户帐户后，作为 Identity Management (IdM)系统管理员，您可能需要使用 SSH 协议连接到远程机器。您可以使用 受限委派 功能来使用 SSH，而无需再次进行身份验证。

按照以下流程，将 Web 控制台配置为使用受限的委托。在以下示例中，web 控制台会话在 myhost.idm.example.com 主机上运行，它被配置为通过代表经过身份验证的用户使用 SSH 访问 remote.idm.example.com 主机。

先决条件

- 已获得 IdM admin ticket-granting ticket (TGT)。
- 有到 remote.idm.example.com 的 root 访问权限。
- web 控制台服务存在于 IdM 中。

- IdM 中存在 remote.idm.example.com 主机。

- web 控制台在用户会话中创建了一个 S4U2Proxy Kerberos ticket。要验证这种情况，以 IdM 用户身份登录 web 控制台，打开 Terminal 页面，输入：

```
$ klist
Ticket cache: FILE:/run/user/1894000001/cockpit-session-3692.ccache
Default principal: user@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
07/30/21 09:19:06 07/31/21 09:19:06
HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
07/30/21 09:19:06  07/31/21 09:19:06  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
        for client HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
```

**流程**

1. 创建可由委派规则访问的目标主机列表：

   a. 创建服务委托目标：

   ```
   $ ipa servicedelegationtarget-add cockpit-target
   ```

   b. 将目标主机添加到委托目标：

   ```
   $ ipa servicedelegationtarget-add-member cockpit-target \ --
   principals=host/remote.idm.example.com@IDM.EXAMPLE.COM
   ```

2. 允许 cockpit 会话通过创建服务委托规则并将 HTTP 服务主体添加到目标主机列表来访问目标主机列表：

   a. 创建服务委托规则：

   ```
   $ ipa servicedelegationrule-add cockpit-delegation
   ```

   b. 将 Web 控制台客户端添加到 delegation 规则中：

   ```
   $ ipa servicedelegationrule-add-member cockpit-delegation \ --
   principals=HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
   ```

-

c. 在委托规则中添加委托目标：

```
$ ipa servicedelegationrule-add-target cockpit-delegation \ --
servicedelegationtargets=cockpit-target
```

3. 在 remote.idm.example.com 主机上启用 Kerberos 身份验证：

a. 以 root 用户身份通过 SSH 连接到 remote.idm.example.com。

b. 打开 /etc/ssh/sshd_config 文件进行编辑。

c. 通过取消注释 GSSAPIAuthentication no 行，并使用 GSSAPIAuthentication yes 替换它来启用 GSSAPIAuthentication。

4. 在 remote.idm.example.com 上重启 SSH 服务，以便上述更改立即生效：

```
$ systemctl try-restart sshd.service
```

**其它资源**

- 使用智能卡登录到 web 控制台
- 身份管理中的受限委托

## 122.3. 使用 ANSIBLE 配置 WEB 控制台，允许用户通过智能卡通过 SSH 向远程主机进行身份验证，而无需再次进行身份验证

在 RHEL web 控制台中登录到用户帐户后，作为 Identity Management (IdM)系统管理员，您可能需要使用 SSH 协议连接到远程机器。您可以使用 受限委派 功能来使用 SSH，而无需再次进行身份验证。

按照以下流程，使用 servicedelegationrule 和 servicedelegationtarget ansible-freeipa 模块将 Web 控制台配置为使用受限的委托。在以下示例中，web 控制台会话在 myhost.idm.example.com 主机上运行，它被配置为通过代表经过身份验证的用户使用 SSH 访问 remote.idm.example.com 主机。

先决条件

- IdM 管理员密码。
- root 访问权限 remote.idm.example.com。
- web 控制台服务存在于 IdM 中。
- IdM 中存在 remote.idm.example.com 主机。
- web 控制台在用户会话中创建了一个 S4U2Proxy Kerberos ticket。要验证这种情况，以 IdM 用户身份登录 web 控制台，打开 Terminal 页面，输入：

```
$ klist
Ticket cache: FILE:/run/user/1894000001/cockpit-session-3692.ccache
Default principal: user@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
07/30/21 09:19:06 07/31/21 09:19:06
HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
07/30/21 09:19:06  07/31/21 09:19:06  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
        for client HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
```

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个具有 IdM 服务器的完全限定域名 (FQDN)的 Ansible 清单文件。
  - 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。
- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 进入您的 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 使用以下内容创建 web-console-smart-card-ssh.yml playbook：

a. 创建确保存在委派目标的任务：

```
---
- name: Playbook to create a constrained delegation target
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure servicedelegationtarget web-console-delegation-target is present
    ipaservicedelegationtarget:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web-console-delegation-target
```

b. 添加将目标主机添加到委托目标的任务：

```
  - name: Ensure servicedelegationtarget web-console-delegation-target member
principal host/remote.idm.example.com@IDM.EXAMPLE.COM is present
    ipaservicedelegationtarget:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web-console-delegation-target
      principal: host/remote.idm.example.com@IDM.EXAMPLE.COM
      action: member
```

c. 添加确保存在委派规则的任务：

```
  - name: Ensure servicedelegationrule delegation-rule is present
    ipaservicedelegationrule:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web-console-delegation-rule
```

d. 添加一个任务，以确保 web 控制台客户端服务的 Kerberos 主体是受限委托规则的成员：

```
  - name: Ensure the Kerberos principal of the web console client service is added
to the servicedelegationrule web-console-delegation-rule
    ipaservicedelegationrule:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web-console-delegation-rule
      principal: HTTP/myhost.idm.example.com
      action: member
```

e. 添加一个任务，以确保受限的委派规则与 web-console-delegation-target delegation 目标关联：

```
  - name: Ensure a constrained delegation rule is associated with a specific
delegation target
    ipaservicedelegationrule:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web-console-delegation-rule
      target: web-console-delegation-target
      action: member
```

3. 保存该文件。

4. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory web-console-
smart-card-ssh.yml
```

5. 在 remote.idm.example.com 中启用 Kerberos 身份验证：

a. 以 root 用户身份通过 SSH 连接到 remote.idm.example.com。

b. 打开 /etc/ssh/sshd_config 文件进行编辑。

c. 通过取消注释 GSSAPIAuthentication no 行，并使用 GSSAPIAuthentication yes 替换它来启用 GSSAPIAuthentication。

**其它资源**

- 使用智能卡登录到 web 控制台
- 身份管理中的受限委托
- README-servicedelegationrule.md 和 README-servicedelegationtarget.md （位于 /usr/share/doc/ansible-freeipa/ 目录中）
- /usr/share/doc/ansible-freeipa/playbooks/servicedelegationtarget 和 /usr/share/doc/ansible-freeipa/playbooks/servicedelegationrule 目录中的 playbook 示例

### 122.4. 配置 WEB 控制台以允许通过智能卡验证的用户运行 SUDO，而无需再次进行身份验证

在登录到 RHEL web 控制台中的用户帐户后，可能需要以 Identity Management (IdM)系统管理员的身份使用超级用户权限运行命令。您可以使用 受限委派 功能在系统中运行 sudo，而无需再次进行身份验证。

按照以下流程，将 Web 控制台配置为使用受限的委托。在以下示例中，web 控制台会话在 myhost.idm.example.com 主机上运行。

**先决条件**

- 已获得 IdM admin ticket-granting ticket (TGT)。
- web 控制台服务存在于 IdM 中。
- IdM 中存在 myhost.idm.example.com 主机。
- 您已为 IdM 服务器 上的域管理员启用了 admin sudo 访问权限。
- web 控制台在用户会话中创建了一个 S4U2Proxy Kerberos ticket。要验证这种情况，以 IdM 用户身份登录 web 控制台，打开 Terminal 页面，输入：

```
$ klist
Ticket cache: FILE:/run/user/1894000001/cockpit-session-3692.ccache
```

```
Default principal: user@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
07/30/21 09:19:06 07/31/21 09:19:06
HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
07/30/21 09:19:06  07/31/21 09:19:06  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
        for client HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
```

**流程**

1. 创建可由委派规则访问的目标主机列表：

   a. 创建服务委托目标：

   ```
   $ ipa servicedelegationtarget-add cockpit-target
   ```

   b. 将目标主机添加到委托目标：

   ```
   $ ipa servicedelegationtarget-add-member cockpit-target \ --
   principals=host/myhost.idm.example.com@IDM.EXAMPLE.COM
   ```

2. 允许 cockpit 会话通过创建服务委托规则并将 HTTP 服务主体添加到目标主机列表来访问目标主机列表：

   a. 创建服务委托规则：

   ```
   $ ipa servicedelegationrule-add cockpit-delegation
   ```

   b. 在 delegation 规则中添加 Web 控制台服务：

   ```
   $ ipa servicedelegationrule-add-member cockpit-delegation \ --
   principals=HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
   ```

   c. 在委托规则中添加委托目标：

   ```
   $ ipa servicedelegationrule-add-target cockpit-delegation \ --
   servicedelegationtargets=cockpit-target
   ```

3. 启用 pam_sss_gss，这是通过通用安全服务应用程序接口(GSSAPI)验证用户身份的 PAM 模

块，并与系统安全服务守护进程(SSSD)协同工作：

a. 打开 /etc/sssd/sssd.conf 文件进行编辑。

b. 指定 pam_sss_gss 可以在 IdM 域中为 sudo 和 sudo -i 命令提供身份验证：

```
[domain/idm.example.com]
pam_gssapi_services = sudo, sudo-i
```

c. 保存并退出文件。

d. 打开 /etc/pam.d/sudo 文件进行编辑。

e. 在 #%PAM-1.0 列表的顶部插入以下行以允许，但不需要 sudo 命令进行 GSSAPI 身份验证：

```
auth sufficient pam_sss_gss.so
```

f. 保存并退出文件。

4. 重启 SSSD 服务，以便上述更改立即生效：

```
$ systemctl restart sssd
```

其它资源

- 使用智能卡登录到 web 控制台
- 身份管理中的受限委托

## 122.5. 使用 ANSIBLE 配置 WEB 控制台，以允许通过智能卡进行身份验证的用户运行 SUDO，而无需再次进行身份验证

在登录到 RHEL web 控制台中的用户帐户后，可能需要以 Identity Management (IdM)系统管理员的

身份使用超级用户权限运行命令。您可以使用 受限委派 功能在系统中运行 sudo，而无需再次进行身份验证。

按照以下流程，使用 ipaservicedelegationrule 和 ipaservicedelegationtarget ansible-freeipa 模块将 Web 控制台配置为使用受限的委托。在以下示例中，web 控制台会话在 myhost.idm.example.com 主机上运行。

先决条件

- 您已通过使用智能卡向 web 控制台会话进行身份验证来获取 IdM admin ticket-granting ticket (TGT)。
- Web 控制台服务已注册到 IdM 。
- IdM 中存在 myhost.idm.example.com 主机。
- 您已为 IdM 服务器 上的域管理员启用了 admin sudo 访问权限。
- web 控制台在用户会话中创建了一个 S4U2Proxy Kerberos ticket。要验证这种情况，以 IdM 用户身份登录 web 控制台，打开 Terminal 页面，输入：

```
$ klist
Ticket cache: FILE:/run/user/1894000001/cockpit-session-3692.ccache
Default principal: user@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
07/30/21 09:19:06 07/31/21 09:19:06
HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
07/30/21 09:19:06  07/31/21 09:19:06  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
        for client HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
```

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包。

- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个 Ansible 清单文件，其中包含您要配置受限委托的 IdM 服务器的完全限定域名(FQDN)。
- 示例假定 secret.yml Ansible 库存储了 ipaadmin_password。

- 目标节点（这是执行 ansible-freeipa 模块的节点）是 IdM 域的一部分，作为 IdM 客户端、服务器或副本的一部分。

**流程**

1. 在 Ansible 控制节点上，导航到 ~/*MyPlaybooks*/ 目录：

```
$ cd ~/MyPlaybooks/
```

2. 使用以下内容创建 web-console-smart-card-sudo.yml playbook：

a. 创建确保存在委派目标的任务：

```
---
- name: Playbook to create a constrained delegation target
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure servicedelegationtarget named sudo-web-console-delegation-target is present
    ipaservicedelegationtarget:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: sudo-web-console-delegation-target
```

b. 添加将目标主机添加到委托目标的任务：

```
  - name: Ensure that a member principal named host/myhost.idm.example.com@IDM.EXAMPLE.COM is present in a service delegation target named sudo-web-console-delegation-target
    ipaservicedelegationtarget:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: sudo-web-console-delegation-target
      principal: host/myhost.idm.example.com@IDM.EXAMPLE.COM
      action: member
```

c. 添加确保存在委派规则的任务：

```
- name: Ensure servicedelegationrule named sudo-web-console-delegation-rule is present
  ipaservicedelegationrule:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: sudo-web-console-delegation-rule
```

d. 添加一个任务，以确保 web 控制台服务的 Kerberos 主体是受限委托规则的成员：

```
- name: Ensure the Kerberos principal of the web console service is added to the service delegation rule named sudo-web-console-delegation-rule
  ipaservicedelegationrule:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: sudo-web-console-delegation-rule
    principal: HTTP/myhost.idm.example.com
    action: member
```

e. 添加一个任务，以确保受限的委派规则与 sudo-web-console-delegation-target delegation 目标关联：

```
- name: Ensure a constrained delegation rule is associated with a specific delegation target
  ipaservicedelegationrule:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: sudo-web-console-delegation-rule
    target: sudo-web-console-delegation-target
    action: member
```

3. 保存该文件。

4. 运行 Ansible playbook。指定 playbook 文件、存储密码的文件保护 secret.yml 文件以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory web-console-smart-card-sudo.yml
```

5. 启用 pam_sss_gss，这是通过通用安全服务应用程序接口(GSSAPI)验证用户身份的 PAM 模块，并与系统安全服务守护进程(SSSD)协同工作：

a. 打开 /etc/sssd/sssd.conf 文件进行编辑。

b. 指定 pam_sss_gss 可以在 IdM 域中为 sudo 和 sudo -i 命令提供身份验证：

```
[domain/idm.example.com]
pam_gssapi_services = sudo, sudo-i
```

c. 保存并退出文件。

d. 打开 /etc/pam.d/sudo 文件进行编辑。

e. 在 #%PAM-1.0 列表的顶部插入以下行以允许，但不需要 sudo 命令进行 GSSAPI 身份验证：

```
auth sufficient pam_sss_gss.so
```

f. 保存并退出文件。

6. 重启 SSSD 服务，以便上述更改立即生效：

```
$ systemctl restart sssd
```

其它资源

- 身份管理中的受限委托
- README-servicedelegationrule.md 和 README-servicedelegationtarget.md （位于 /usr/share/doc/ansible-freeipa/ 目录中）
- /usr/share/doc/ansible-freeipa/playbooks/servicedelegationtarget 和 /usr/share/doc/ansible-freeipa/playbooks/servicedelegationrule 目录中的 playbook 示例

### 122.6. 其它资源

- 在 web 控制台中管理远程系统