# Red Hat Enterprise Linux 9

## 管理 IdM 用户、组、主机和访问控制规则

配置用户和主机，在组中管理它们，并使用基于主机和基于角色的访问控制规则控制访问

Last Updated: 2024-06-27

# Red Hat Enterprise Linux 9 管理 IdM 用户、组、主机和访问控制规则

配置用户和主机，在组中管理它们，并使用基于主机和基于角色的访问控制规则控制访问

## 法律通告



## 摘要

红帽身份管理(IdM)的主要功能是管理用户、组、主机和访问控制规则，如基于主机的访问控制(HBAC)和基于角色的访问控制(RBAC)。您可以使用命令行、IdM Web UI 和 Ansible Playbook 配置它们。 管理任务包括配置 Kerberos 策略和安全性、自动化组成员资格和委派权限。

# 目录

**使开源包含更多** ..... 11

对红帽文档提供反馈 ..... 12

**第 1 章 IDM 命令行工具简介** ..... 13
1.1. 什么是 IPA 命令行界面 13
1.2. IPA 帮助是什么 13
1.3. 使用 IPA 帮助主题 14
1.4. 使用 IPA HELP 命令 14
1.5. IPA 命令的结构 15
1.6. 使用 IPA 命令将用户帐户添加到 IDM 15
1.7. 使用 IPA 命令修改 IDM 中的用户帐户 17
1.8. 如何为 IDM 工具提供值列表 17
1.9. 如何在 IDM 工具中使用特殊字符 18

**第 2 章 使用命令行管理用户帐户** ..... 19
2.1. 用户生命周期 19
2.2. 使用命令行添加用户 20
2.3. 使用命令行激活用户 21
2.4. 使用命令行保留用户 22
2.5. 使用命令行删除用户 22
2.6. 使用命令行恢复用户 23

**第 3 章 使用 IDM WEB UI 管理用户帐户** ..... 24
3.1. 用户生命周期 24
3.2. 在 WEB UI 中添加用户 25
3.3. 在 IDM WEB UI 中 STAGE 用户 27
3.4. 在 WEB UI 中禁用用户帐户 28
3.5. 在 WEB UI 中启用用户帐户 29
3.6. 在 IDM WEB UI 中保留活动的用户 30
3.7. 在 IDM WEB UI 中恢复用户 31
3.8. 在 IDM WEB UI 中删除用户 31

**第 4 章 使用 ANSIBLE PLAYBOOK 管理用户帐户** ..... 33
4.1. 用户生命周期 33
4.2. 使用 ANSIBLE PLAYBOOK 确保存在一个 IDM 用户 34
4.3. 使用 ANSIBLE PLAYBOOK 确保存在多个 IDM 用户 36
4.4. 使用 ANSIBLE PLAYBOOK 确保存在 JSON 文件中的多个 IDM 用户 38
4.5. 确保没有用户使用 ANSIBLE PLAYBOOK 39
4.6. 其他资源 41

**第 5 章 在 IDM 中管理用户密码** ..... 42
5.1. 谁可以更改 IDM 用户密码以及如何去做 42
5.2. 在 IDM WEB UI 中更改用户密码 42
5.3. 在 IDM WEB UI 中重置另一个用户的密码 42
5.4. 重置目录管理器用户密码 43
5.5. 在 IDM CLI 中更改您的用户密码或重置另一个用户的密码 44
5.6. 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 44
5.7. 检查 IDM 用户帐户是否已被锁住 46
5.8. 在 IDM 中密码失败后解锁用户帐户 47
5.9. 为 IDM 中的用户启用最后一次成功 KERBEROS 验证的跟踪 47

**第 6 章 定义 IDM 密码策略** ..... 49

6.1. 什么是密码策略 49
6.2. IDM 中的密码策略 49
6.3. 使用 ANSIBLE PLAYBOOK 在 IDM 中存在密码策略 50
6.4. IDM 中的附加密码策略选项 52
6.5. 将其他密码策略选项应用到 IDM 组 53
6.6. 使用 ANSIBLE PLAYBOOK 将额外的密码策略选项应用到 IDM 组 55

**第 7 章 管理过期密码通知 59**
7.1. 什么是过期的密码通知工具 59
7.2. 安装过期的密码通知工具 59
7.3. 运行 EPN 工具，向密码即将过期的用户发送电子邮件 59
7.4. 启用 IPA-EPN.TIMER，向密码即将过期的所有用户发送电子邮件 61
7.5. 修改过期密码通知电子邮件模板 62

**第 8 章 为 IDM 客户端上的 IDM 用户授予 SUDO 访问权限 64**
8.1. IDM 客户端上的 SUDO 访问权限 64
8.2. 使用 CLI 向 IDM 客户端上的 IDM 用户授予 SUDO 访问权限 64
8.3. 使用 CLI 向 IDM 客户端上的 AD 用户授予 SUDO 访问权限 66
8.4. 使用 IDM WEB UI 向 IDM 客户端上的 IDM 用户授予 SUDO 访问权限 70
8.5. 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 72
8.6. 在 IDM WEBUI 中创建一个 SUDO 规则，该规则以 IDM 客户端上服务帐户的身份运行命令 75
8.7. 在 IDM 客户端上为 SUDO 启用 GSSAPI 身份验证 80
8.8. 在 IDM 客户端上启用 GSSAPI 身份验证，并为 SUDO 强制使用 KERBEROS 身份验证指示符 82
8.9. SSSD 选项控制 PAM 服务的 GSSAPI 身份验证 84
8.10. SUDO 的 GSSAPI 身份验证故障排除 85
8.11. 使用 ANSIBLE PLAYBOOK 来确保 IDM 客户端上 IDM 用户的 SUDO 访问权限 87

**第 9 章 使用 LDAPMODIFY 在外部管理 IDM 用户 90**
9.1. 在外部管理 IDM 用户帐户的模板 90
9.2. 在外部管理 IDM 组帐户的模板 92
9.3. 以互动方式使用 LDAPMODIFY 命令 93
9.4. 使用 LDAPMODIFY 保留 IDM 用户 94

**第 10 章 使用 LDAPSEARCH 命令搜索 IDM 条目 96**
10.1. 使用 LDAPSEARCH 命令 96
10.2. 使用 LDAPSEARCH 过滤器 97

**第 11 章 为用户的外部调配配置 IDM 99**
11.1. 为 STAGE 用户帐户的自动激活准备 IDM 帐户 99
11.2. 配置 IDM STAGE用户帐户的自动激活 101
11.3. 添加 LDIF 文件中定义的 IDM STAGE 用户 102
11.4. 使用 LDAPMODIFY 直接从 CLI 添加 IDM STAGE 用户 104
11.5. 其他资源 106

**第 12 章 为用户、主机和服务管理 KERBEROS 主体别名 107**
12.1. 添加一个 KERBEROS 主体别名 107
12.2. 删除一个 KERBEROS 主体别名 107
12.3. 添加一个 KERBEROS 企业主体别名 108
12.4. 删除 KERBEROS 企业主体别名 108

**第 13 章 使用 PAC 信息增强 KERBEROS 安全性 110**
13.1. IDM 中使用特权属性证书 (PAC) 110
13.2. 在 IDM 中启用安全标识符 (SID) 110

**第 14 章 管理 KERBEROS 票据策略 112**

14.1. IDM KDC 的角色 112
14.2. IDM KERBEROS 票据策略类型 113
14.3. KERBEROS 认证指示符 114
14.4. 为 IDM 服务强制执行身份验证指标 115
14.5. 配置全局票据生命周期策略 120
14.6. 根据身份验证指标配置全局票据策略 121
14.7. 为用户配置默认的票据策略 122
14.8. 为用户配置单独的身份验证指标票据策略 122
14.9. KRBTPOLICY-MOD 命令的身份验证指标选项 123

**第 15 章 IDM 中的 KERBEROS PKINIT 身份验证 125**
15.1. 默认 PKINIT 配置 125
15.2. 显示当前 PKINIT 配置 125
15.3. 在 IDM 中配置 PKINIT 126
15.4. 其他资源 127

**第 16 章 维护 IDM KERBEROS KEYTAB 文件 128**
16.1. IDENTITY MANAGEMENT 如何使用 KERBEROS KEYTAB 文件 128
16.2. 验证 KERBEROS KEYTAB 文件是否与 IDM 数据库同步 129
16.3. IDM KERBEROS KEYTAB 文件内容列表 130
16.4. 查看 IDM 主密钥的加密类型 131

**第 17 章 在 IDM 环境中启用 PASSKEY 身份验证 132**
17.1. 先决条件 132
17.2. 注册 PASSKEY 设备 132
17.3. 身份验证策略 133
17.4. 以 PASSKEY 用户身份检索 IDM TICKET-GRANTING TICKET 134

**第 18 章 在 IDM 中使用 KDC 代理 136**
18.1. 配置 IDM 客户端以使用 KKDCP 136
18.2. 验证 IDM 服务器上是否启用了 KKDCP 136
18.3. 在 IDM 服务器上禁用 KKDCP 137
18.4. 在 IDM 服务器上重新启用 KKDCP 137
18.5. 配置 KKDCP 服务器 I 138
18.6. 配置 KKDCP 服务器 II 139

**第 19 章 使用 CLI 管理 IDM 中的自助服务规则 140**
19.1. IDM 中的自助服务访问控制 140
19.2. 使用 CLI 创建自助服务规则 140
19.3. 使用 CLI 编辑自助服务规则 141
19.4. 使用 CLI 删除自助服务规则 141

**第 20 章 使用 IDM WEB UI 管理自助服务规则 143**
20.1. IDM 中的自助服务访问控制 143
20.2. 使用 IDM WEB UI 创建自助服务规则 143
20.3. 使用 IDM WEB UI 编辑自助服务规则 145
20.4. 使用 IDM WEB UI 删除自助服务规则 146

**第 21 章 使用 ANSIBLE PLAYBOOK 管理 IDM 中的自助服务规则 147**
21.1. IDM 中的自助服务访问控制 147
21.2. 使用 ANSIBLE 确保存在自助服务规则 147
21.3. 使用 ANSIBLE 确保缺少自助服务规则 149
21.4. 使用 ANSIBLE 确保自助服务规则具有特定属性 150
21.5. 使用 ANSIBLE 确保自助服务规则没有特定属性 152

**第 22 章 在 IDM CLI 中管理用户组** ..... **154**
22.1. IDM 中的不同组类型 154
22.2. 直接和间接组成员 155
22.3. 使用 IDM CLI 添加用户组 155
22.4. 使用 IDM CLI 搜索用户组 156
22.5. 使用 IDM CLI 删除用户组 156
22.6. 使用 IDM CLI 将成员添加到用户组中 156
22.7. 添加没有用户私有组的用户 157
22.8. 使用 IDM CLI 将用户或组作为成员管理者添加到 IDM 用户组中 159
22.9. 使用 IDM CLI 查看组成员 160
22.10. 使用 IDM CLI 从用户组中删除成员 161
22.11. 使用 IDM CLI 从 IDM 用户组中删除作为成员管理者的用户或组 161
22.12. 为 IDM 中的本地和远程组启用组合并 162
22.13. 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 164

**第 23 章 在 IDM WEB UI 中管理用户组** ..... **167**
23.1. IDM 中的不同组类型 167
23.2. 直接和间接组成员 169
23.3. 使用 IDM WEB UI 添加用户组 169
23.4. 使用 IDM WEB UI 删除用户组 170
23.5. 使用 IDM WEB UI 将成员添加到用户组中 171
23.6. 使用 WEB UI 将用户或组作为成员管理者添加到 IDM 用户组中 172
23.7. 使用 IDM WEB UI 查看组成员 175
23.8. 使用 IDM WEB UI 从用户组中删除成员 176
23.9. 使用 WEB UI 从 IDM 用户组中删除作为成员管理者的用户或组 177

**第 24 章 使用 ANSIBLE PLAYBOOK 管理用户组** ..... **179**
24.1. IDM 中的不同组类型 179
24.2. 直接和间接组成员 181
24.3. 使用 ANSIBLE PLAYBOOK 确保存在 IDM 组和组成员 182
24.4. 使用 ANSIBLE 在单个任务中添加多个 IDM 组 184
24.5. 使用 ANSIBLE 启用 AD 用户管理 IDM 186
24.6. 使用 ANSIBLE PLAYBOOK 确保 IDM 用户组中存在成员管理器 188
24.7. 使用 ANSIBLE PLAYBOOK 确保 IDM 用户组中没有成员管理者 190

**第 25 章 使用 IDM CLI 自动化组成员资格** ..... **193**
25.1. 自动化组成员资格的好处 194
25.2. 自动成员规则 194
25.3. 使用 IDM CLI 添加自动成员规则 195
25.4. 使用 IDM CLI 将条件添加到自动成员规则中 196
25.5. 使用 IDM CLI 查看现有的自动成员规则 198
25.6. 使用 IDM CLI 删除自动成员规则 199
25.7. 使用 IDM CLI 从自动成员规则中删除条件 199
25.8. 使用 IDM CLI 将自动成员规则应用到现有条目 200
25.9. 使用 IDM CLI 配置默认的自动成员组 201

**第 26 章 使用 IDM WEB UI 自动化组成员资格** ..... **204**
26.1. 自动化组成员资格的好处 205
26.2. 自动成员规则 205
26.3. 使用 IDM WEB UI 添加自动成员规则 206
26.4. 使用 IDM WEB UI 向自动成员规则中添加条件 207
26.5. 使用 IDM WEB UI 查看现有的自动成员规则和条件 209
26.6. 使用 IDM WEB UI 删除自动成员规则 210
26.7. 使用 IDM WEB UI 从自动成员规则中删除条件 211

26.8. 使用 IDM WEB UI 将自动成员规则应用到现有条目 212
26.9. 使用 IDM WEB UI 配置默认的用户组 214
26.10. 使用 IDM WEB UI 配置默认的主机组 215

**第 27 章 使用 ANSIBLE 在 IDM 中自动化组成员资格 ..... 217**
27.1. 准备 ANSIBLE 控制节点来管理 IDM 217
27.2. 使用 ANSIBLE 确保 IDM 用户组的自动成员规则存在 220
27.3. 使用 ANSIBLE 确保指定的条件在 IDM 用户组自动成员规则中存在 222
27.4. 使用 ANSIBLE 确保条件在 IDM 用户组自动成员规则中不存在 226
27.5. 使用 ANSIBLE 确保 IDM 用户组的自动成员规则不存在 229
27.6. 使用 ANSIBLE 确保 IDM 主机组自动成员规则中存在条件 231
27.7. 其他资源 234

**第 28 章 将权限委派给用户组，来使用 IDM CLI 管理用户 ..... 235**
28.1. 委派规则 235
28.2. 使用 IDM CLI 创建委派规则 235
28.3. 使用 IDM CLI 查看现有的委派规则 236
28.4. 使用 IDM CLI 修改委派规则 237
28.5. 使用 IDM CLI 删除委派规则 238

**第 29 章 将权限委派给用户组，来使用 IDM WEB UI 管理用户 ..... 239**
29.1. 委派规则 239
29.2. 使用 IDM WEBUI 创建委派规则 239
29.3. 使用 IDM WEBUI 查看现有的委派规则 241
29.4. 使用 IDM WEBUI 修改委派规则 242
29.5. 使用 IDM WEBUI 删除委派规则 244

**第 30 章 委派权限到用户组，以使用 ANSIBLE PLAYBOOK 管理用户 ..... 245**
30.1. 委派规则 245
30.2. 为 IDM 创建 ANSIBLE 清单文件 245
30.3. 使用 ANSIBLE 确保存在委派规则 247
30.4. 使用 ANSIBLE 确保没有委派规则 249
30.5. 使用 ANSIBLE 确保委派规则具有特定属性 252
30.6. 使用 ANSIBLE 确保委派规则没有特定属性 254

**第 31 章 使用 CLI 在 IDM 中管理基于角色的访问控制 ..... 257**
31.1. IDM 中的基于角色的访问控制 257
31.2. 在 CLI 中管理 IDM 权限 263
31.3. 现有权限的命令选项 266
31.4. 在 CLI 中管理 IDM 特权 266
31.5. 现有权限的命令选项 267
31.6. 在 CLI 中管理 IDM 角色 268
31.7. 现有角色的命令选项 269

**第 32 章 使用 IDM WEB UI 管理基于角色的访问控制 ..... 270**
32.1. IDM 中的基于角色的访问控制 270
32.2. 在 IDM WEB UI 中管理权限 276
32.3. 在 IDM WEB UI 中管理特权 281
32.4. 在 IDM WEB UI 中管理角色 284

**第 33 章 准备您的环境以使用 ANSIBLE PLAYBOOK 管理 IDM ..... 290**

**第 34 章 在 IDM 中使用 ANSIBLE PLAYBOOK 管理基于角色的访问控制 ..... 293**
34.1. IDM 中的权限 294
34.2. 默认管理的权限 295

34.3. IDM 中的特权 298
34.4. IDM 中的角色 298
34.5. IDENTITY MANAGEMENT 中的预定义角色 299
34.6. 使用 ANSIBLE 确保存在带有特权的 IDM RBAC 角色 299
34.7. 使用 ANSIBLE 确保缺少 IDM RBAC 角色 302
34.8. 使用 ANSIBLE 确保为一组用户分配 IDM RBAC 角色 304
34.9. 使用 ANSIBLE 确保没有将特定用户分配给 IDM RBAC 角色 307
34.10. 使用 ANSIBLE 确保服务是 IDM RBAC 角色的成员 309
34.11. 使用 ANSIBLE 确保主机是 IDM RBAC 角色的成员 312
34.12. 使用 ANSIBLE 确保主机组是 IDM RBAC 角色的成员 314

**第 35 章 使用 ANSIBLE PLAYBOOK 管理 RBAC 特权 318**
35.1. 使用 ANSIBLE 确保存在自定义 IDM RBAC 特权 318
35.2. 使用 ANSIBLE 确保自定义 IDM RBAC 特权中存在成员权限 320
35.3. 使用 ANSIBLE 确保 IDM RBAC 特权不包括权限 323
35.4. 使用 ANSIBLE 重命名自定义 IDM RBAC 特权 325
35.5. 使用 ANSIBLE 确保缺少 IDM RBAC 特权 328
35.6. 其他资源 330

**第 36 章 使用 ANSIBLE PLAYBOOK 在 IDM 中管理 RBAC 权限 331**
36.1. 使用 ANSIBLE 确保存在 RBAC 权限 331
36.2. 使用 ANSIBLE 确保存在带有属性的 RBAC 权限 334
36.3. 使用 ANSIBLE 确保缺少 RBAC 权限 337
36.4. 使用 ANSIBLE 确保属性是 IDM RBAC 权限的成员 339
36.5. 使用 ANSIBLE 确保属性不是 IDM RBAC 权限的成员 342
36.6. 使用 ANSIBLE 重命名 IDM RBAC 权限 344
36.7. 其他资源 346

**第 37 章 使用 ID 视图来覆盖 IDM 客户端上的用户属性值 348**
37.1. ID 视图 348
37.2. ID 视图对 SSSD 性能的潜在负面影响 349
37.3. ID 视图可以覆盖的属性 349
37.4. 获取 ID 视图命令的帮助信息 350
37.5. 使用 ID 视图来覆盖特定主机上 IDM 用户的登录名称 351
37.6. 修改 IDM ID 视图 354
37.7. 添加 ID 视图来覆盖 IDM 客户端上的 IDM 用户主目录 356
37.8. 将 ID 视图应用到 IDM 主机组 359
37.9. 使用 ANSIBLE 覆盖特定主机上 IDM 用户的登录名称和主目录 362
37.10. 使用 ANSIBLE 配置在 IDM 客户端上启用 SSH 密钥登录的 ID 视图 364
37.11. 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 367
37.12. 使用 ANSIBLE 确保带有特定 UID 的 ID 视图中存在 IDM 用户 369
37.13. 使用 ANSIBLE 确保 IDM 用户可以使用两个证书登录到 IDM 客户端 371
37.14. 使用 ANSIBLE 为 IDM 客户端上的声音卡授予 IDM 组访问权限 373
37.15. 将 NIS 域迁移到身份管理 376

**第 38 章 为活动目录用户使用 ID 视图 378**
38.1. DEFAULT TRUST VIEW 是如何工作的 378
38.2. 通过修改 DEFAULT TRUST VIEW 为 AD 用户定义全局属性 379
38.3. 对带有 ID 视图的 IDM 客户端上的 AD 用户覆盖 DEFAULT TRUST VIEW 属性 380
38.4. 将 ID 视图应用到 IDM 主机组 382

**第 39 章 手动调整 ID 范围 386**
39.1. ID 范围 386
39.2. 自动 ID 范围分配 387

39.3. 在服务器安装过程中手动分配 IDM ID 范围 387
39.4. 添加新的 IDM ID 范围 388
39.5. IDM ID 范围中的安全性和相对标识符的角色 390
39.6. 使用 ANSIBLE 添加新的本地 IDM ID 范围 392
39.7. 删除对 AD 的信任后删除 ID 范围 395
39.8. 显示当前分配的 DNA ID 范围 396
39.9. 手动 ID 范围分配 397
39.10. 手动分配 DNA ID 范围 398

**第 40 章 手动管理 SUBID 范围 400**
40.1. 使用 IDM CLI 生成子 SUBID 范围 400
40.2. 使用 IDM WEBUI 接口生成 SUBID 范围 401
40.3. 使用 IDM CLI 查看有关 IDM 用户的 SUBID 信息 402
40.4. 使用 GETSUBID 命令列出 SUBID 范围 403

**第 41 章 在 IDM CLI 中管理主机 405**
41.1. IDM 中的主机 405
41.2. 主机注册 406
41.3. 主机注册所需的用户权限 407
41.4. IDM 主机和用户的注册和身份验证：比较 408
41.5. 主机操作 409
41.6. IDM LDAP 中的主机条目 411
41.7. 从 IDM CLI 添加 IDM 主机条目 413
41.8. 从 IDM CLI 删除主机条目 414
41.9. 重新注册身份管理客户端 414
41.10. 重命名身份管理客户端系统 416
41.11. 禁用和重新启用主机条目 419

**第 42 章 从 IDM WEB UI 添加主机条目 422**
42.1. IDM 中的主机 422
42.2. 主机注册 423
42.3. 主机注册所需的用户权限 423
42.4. IDM 主机和用户的注册和身份验证：比较 424
42.5. IDM LDAP 中的主机条目 426
42.6. 从 WEB UI 添加主机条目 427

**第 43 章 使用 ANSIBLE PLAYBOOK 管理主机 430**
43.1. 使用 ANSIBLE PLAYBOOK 确保存在带有 FQDN 的 IDM 主机条目 430
43.2. 使用 ANSIBLE PLAYBOOK 确保存在含有 DNS 信息的 IDM 主机条目 433
43.3. 使用 ANSIBLE PLAYBOOK 确保存在带有随机密码的多个 IDM 主机条目 435
43.4. 使用 ANSIBLE PLAYBOOK 确保存在具有多个 IP 地址的 IDM 主机条目 437
43.5. 使用 ANSIBLE PLAYBOOK 确保没有 IDM 主机条目 440
43.6. 其他资源 442

**第 44 章 使用 IDM CLI 管理主机组 443**
44.1. IDM 中的主机组 443
44.2. 使用 CLI 查看 IDM 主机组 444
44.3. 使用 CLI 创建 IDM 主机组 445
44.4. 使用 CLI 删除 IDM 主机组 445
44.5. 使用 CLI 添加 IDM 主机组成员 446
44.6. 使用 CLI 删除 IDM 主机组成员 447
44.7. 使用 CLI 添加 IDM 主机组成员管理者 449
44.8. 使用 CLI 删除 IDM 主机组成员管理者 450

**第 45 章 使用 IDM WEB UI 管理主机组 453**

45.1. IDM 中的主机组 453
45.2. 在 IDM WEB UI 中查看主机组 454
45.3. 在 IDM WEB UI 中创建主机组 455
45.4. 在 IDM WEB UI 中删除主机组 456
45.5. 在 IDM WEB UI 中添加主机组成员 457
45.6. 在 IDM WEB UI 中删除主机组成员 458
45.7. 使用 WEB UI 添加 IDM 主机组成员管理者 459
45.8. 使用 WEB UI 删除 IDM 主机组成员管理者 461

**第 46 章 使用 ANSIBLE PLAYBOOK 管理主机组 464**
46.1. IDM 中的主机组 464
46.2. 使用 ANSIBLE PLAYBOOK 确保存在 IDM 主机组 465
46.3. 确保使用 ANSIBLE PLAYBOOK 在 IDM 主机组中存在主机 467
46.4. 使用 ANSIBLE PLAYBOOK 嵌套 IDM 主机组 469
46.5. 使用 ANSIBLE PLAYBOOK 在 IDM 主机组中存在成员管理器 471
46.6. 使用 ANSIBLE PLAYBOOK 确保 IDM 主机组中没有主机 474
46.7. 使用 ANSIBLE PLAYBOOK 确保 IDM 主机组没有嵌套的主机组 476
46.8. 使用 ANSIBLE PLAYBOOK 确保没有 IDM 主机组 478
46.9. 使用 ANSIBLE PLAYBOOK 确保 IDM 主机组中没有成员管理器 480

**第 47 章 配置基于主机的访问控制规则 483**
47.1. 使用 WEBUI 在 IDM 域中配置 HBAC 规则 483
47.2. 使用 CLI 在 IDM 域中配置 HBAC 规则 487
47.3. 为自定义 HBAC 服务添加 HBAC 服务条目 492
47.4. 添加 HBAC 服务组 493

**第 48 章 使用 ANSIBLE PLAYBOOK 确保在 IDM 中存在基于主机的访问控制规则 495**
48.1. IDM 中的基于主机的访问控制规则 495
48.2. 使用 ANSIBLE PLAYBOOK 确保在 IDM 中存在 HBAC 规则 495

**第 49 章 管理用户和主机的公共 SSH 密钥 498**
49.1. 关于 SSH 密钥格式 498
49.2. 关于 IDM 和 OPENSSH 499
49.3. 生成 SSH 密钥 500
49.4. 管理主机的公用 SSH 密钥 501
49.5. 管理用户的公共 SSH 密钥 505

**第 50 章 配置域解析顺序来解析简短的 AD 用户名 510**
50.1. 域解析顺序的工作方式 510
50.2. 在 IDM 服务器上设置全局域解析顺序 511
50.3. 为 IDM 服务器上的 ID 视图设置域解析顺序 512
50.4. 使用 ANSIBLE 创建 ID 视图，其域解析顺序 514
50.5. 在 IDM 客户端上的 SSSD 中设置域解析顺序 516
50.6. 其他资源 517

**第 51 章 在 IDM 中使用 AD USER PRINCIPAL NAMES 启用身份验证 518**
51.1. IDM 信任的 AD 林中的用户主体名称 518
51.2. 确保 AD UPN 在 IDM 中是最新的 519
51.3. 为 AD UPN 身份验证问题收集故障排除数据 520

**第 52 章 启用 AD 用户管理 IDM 522**
52.1. AD 用户的 ID 覆盖 522
52.2. 使用 ID 覆盖来启用 AD 用户管理 IDM 522
52.3. 使用 ANSIBLE 启用 AD 用户管理 IDM 523
52.4. 验证 AD 用户是否可以在 IDM CLI 中执行正确的命令 526

52.5. 使用 ANSIBLE 启用 AD 用户管理 IDM 526

**第 53 章 使用外部身份提供程序向 IDM 进行身份验证 ..... 530**
53.1. 将 IDM 连接到外部 IDP 的好处 530
53.2. IDM 如何通过外部 IDP 融合登录 530
53.3. 创建对外部身份提供程序的引用 532
53.4. IDM 中不同外部 IDP 的引用示例 533
53.5. 在 IDM 中管理外部身份提供程序的 IPA IDP114 命令的选项 534
53.6. 管理对外部 IDP 的引用 536
53.7. 启用 IDM 用户通过外部 IDP 进行身份验证 537
53.8. 以外部 IDP 用户身份检索 IDM TICKET-GRANTING TICKET 538
53.9. 以外部 IDP 用户身份通过 SSH 登录到 IDM 客户端 540
53.10. IPA IDP114 命令中的 --PROVIDER 选项 541

**第 54 章 使用 ANSIBLE 将 IDM 用户的身份验证委派给外部身份提供程序 ..... 546**
54.1. 将 IDM 连接到外部 IDP 的好处 546
54.2. IDM 如何通过外部 IDP 融合登录 546
54.3. 使用 ANSIBLE 创建对外部身份提供程序的引用 547
54.4. 使用 ANSIBLE 启用 IDM 用户通过外部 IDP 进行身份验证 549
54.5. 以外部 IDP 用户身份检索 IDM TICKET-GRANTING TICKET 552
54.6. 以外部 IDP 用户身份通过 SSH 登录到 IDM 客户端 554
54.7. IPAIDP ANSIBLE 模块中的 PROVIDER 选项 555

**第 55 章 在 IDM 中使用基于资源的受限委托 ..... 560**
55.1. 其他资源 560
55.2. 在身份管理中基于资源的受限委托 560
55.3. 使用 RBCD 委派对服务的访问 561

# 使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 CTO Chris Wright 的信息。

# 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. 点顶部导航栏中的 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您的改进建议。包括到文档相关部分的链接。
5. 点对话框底部的 **Create**。

# 第 1 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，请通过有效的 Kerberos 票据向 IdM 进行身份验证。

## 1.1. 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。

它支持很多用于管理 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。

## 1.2. 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。

执行不带选项的 **ipa help** 命令可显示有关基本帮助用法和最常见的命令示例的信息。

您可以将以下选项用于不同的 **ipa help** 用例：

```
$ 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 **ipa help** 命令指定 **TOPIC**、**COMMAND** 或 **topics**、或 **commands** ：
  - **topics**-- 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 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** -- 使用 **ipa help commands** 以显示 IPA 帮助命令涵盖的命令列表，如 **user-add**、**ca-enable**、**server-show** 等。
- **COMMAND** -- 大写 COMMAND 是一个变量。因此，您可以指定特定的命令，例如 **ipa help user-add**。

## 1.3. 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，再连接到 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. 选择其中一个主题并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 help 的输出太长，且您无法看到整个文本，请使用以下语法：

```
$ 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。

## 1.4. 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，再连接到 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. 选择一个命令并根据以下模式创建 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
```

**其他资源**

- **ipa** 手册页。

## 1.5. IPA 命令的结构

IPA CLI 区分以下命令类型：

- **内置命令** - IdM 服务器中提供了所有内置命令。
- **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。

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

## 1.6. 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添加新用户到 Identity Management (IdM) 数据库。

**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，再连接到 IdM 服务器。
2. 输入命令来添加新用户：

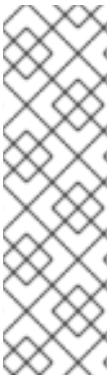
```
$ ipa user-add
```

该命令将运行一个脚本，用于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]:**中输入用户名，或者按 **Enter** 键接受推荐的用户名。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了错误，使用 **ipa user-add** 命令重复该过程，并使用一个不同的唯一用户名。

添加用户名后，用户帐户将添加到 IdM 数据库，IPA 命令行界面 (CLI) 会输出以下内容：

```
----------------------
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，使用以下语法运行 **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。

**其他资源**

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

## 1.7. 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，再连接到 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 **--password** 来添加密码：

   ```
   $ 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 输出以下内容：

```
----------------------
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。

**其他资源**

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

## 1.8. 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**，其为对象添加权限。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，在上面的命令中，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} ...
```

## 1.9. 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 2 章 使用命令行管理用户帐户

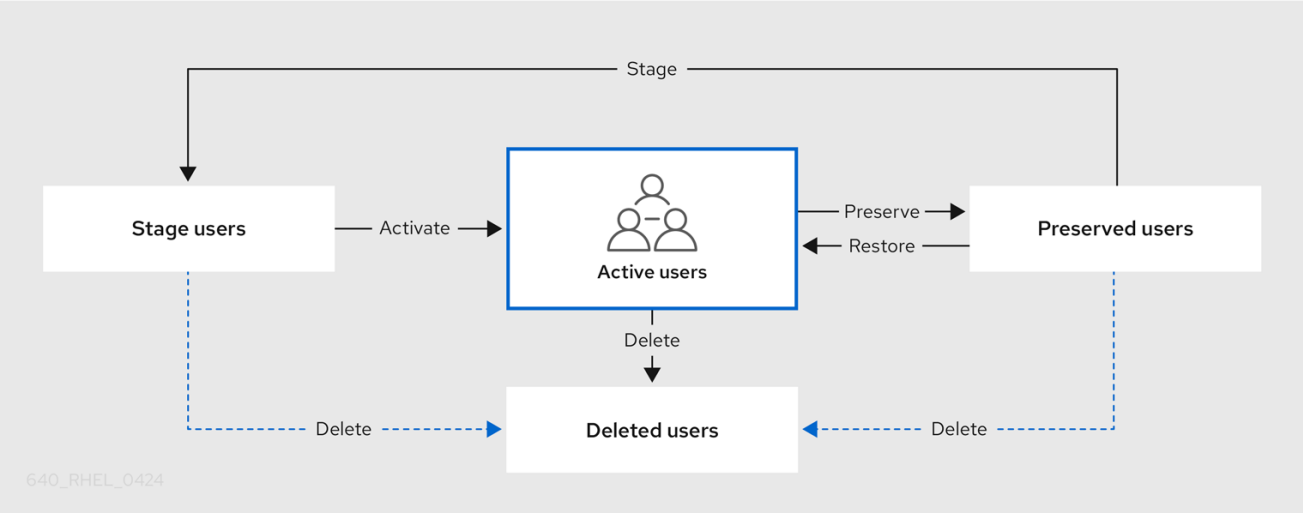
IdM （身份管理）的用户生命周期中有几个阶段，包括：

- 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

## 2.1. 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。

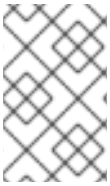
## 2.2. 使用命令行添加用户

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。

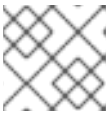
**步骤**

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。

## 2.3. 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。

**步骤**

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。

## 2.4. 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。

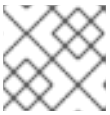
**先决条件**

- 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。

**步骤**

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。

## 2.5. 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。

**步骤**

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。

## 2.6. 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。

**步骤**

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 3 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。

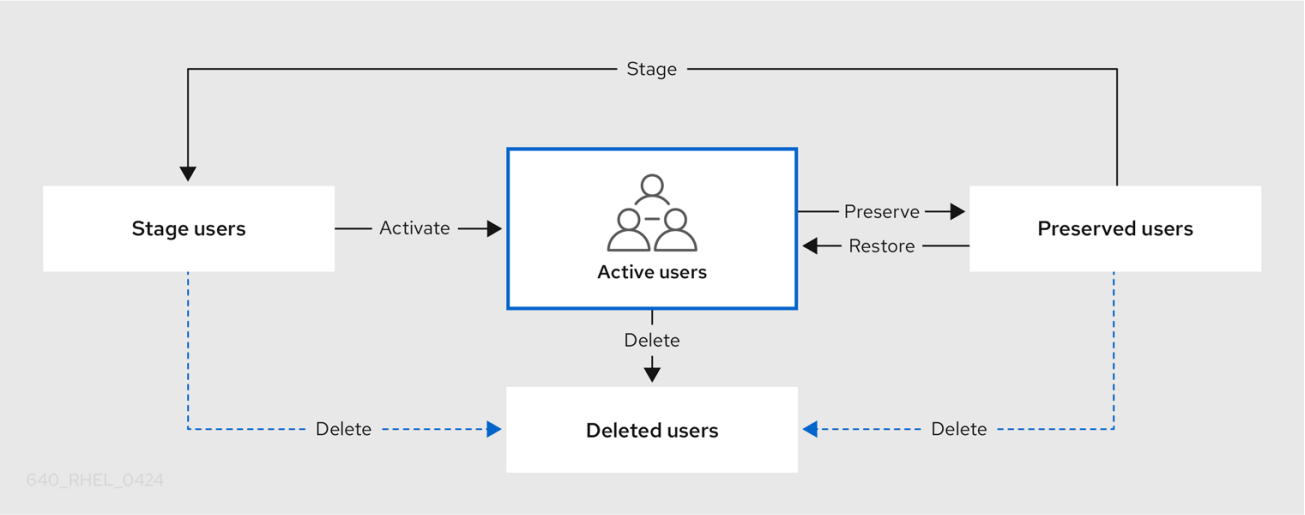
**删除用户帐户**

如果员工离职，在不需要备份的情况下删除该帐户。

## 3.1. 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。

## 3.2. 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**

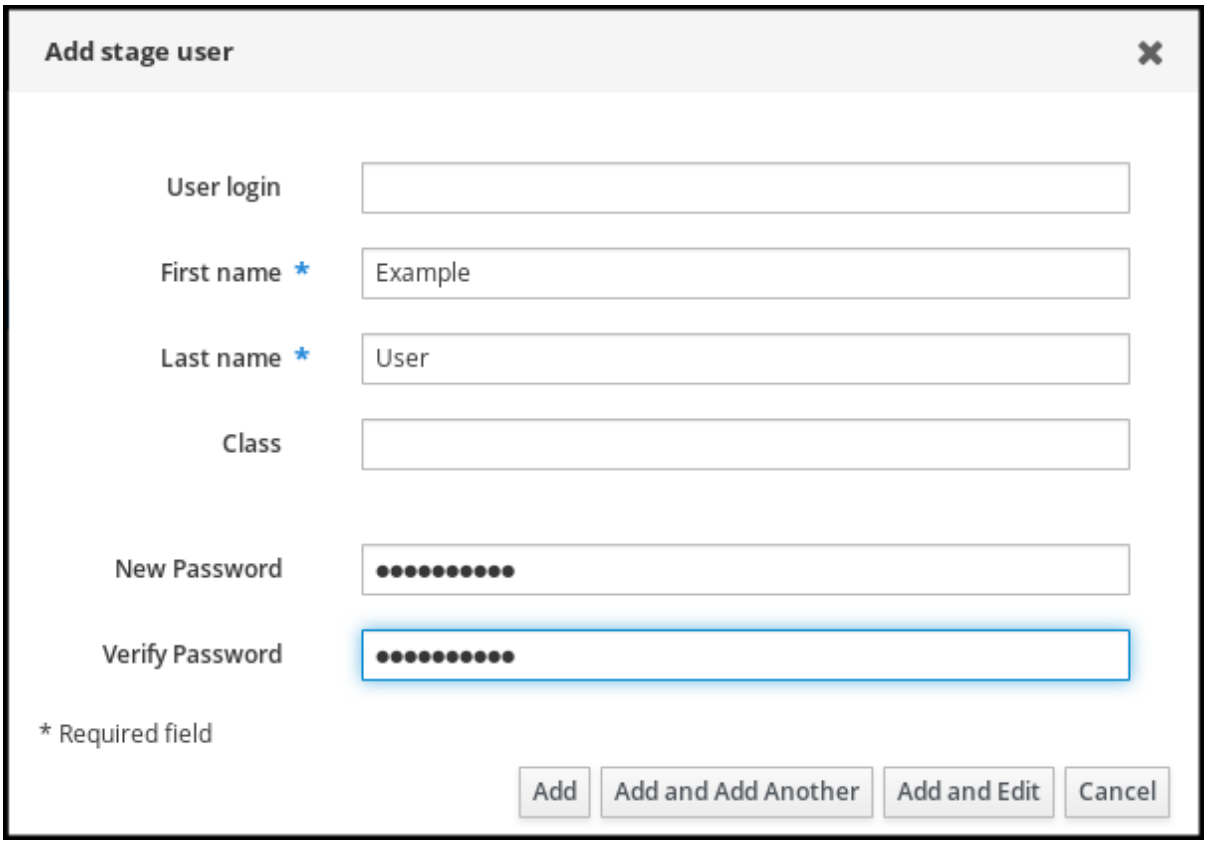
或者，您可以直接创建活动的用户帐户。要添加活动的用户，请按照下面的流程，并在 **Active users** 选项卡中添加用户帐户。

**先决条件**

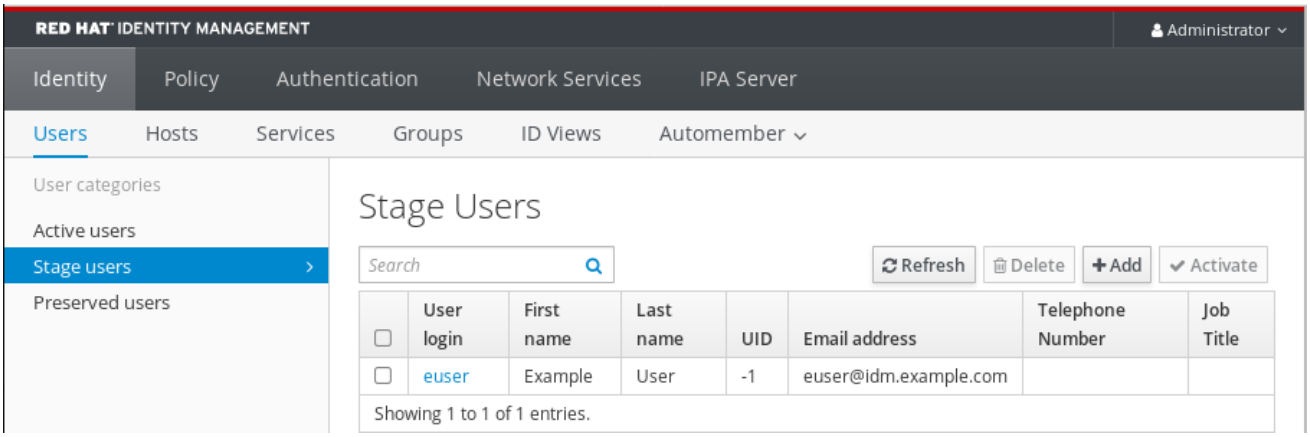
- 管理 IdM 或用户管理员角色的管理员特权。

**步骤**

1. 登录到 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。

## 3.3. 在 IDM WEB UI 中 STAGE 用户

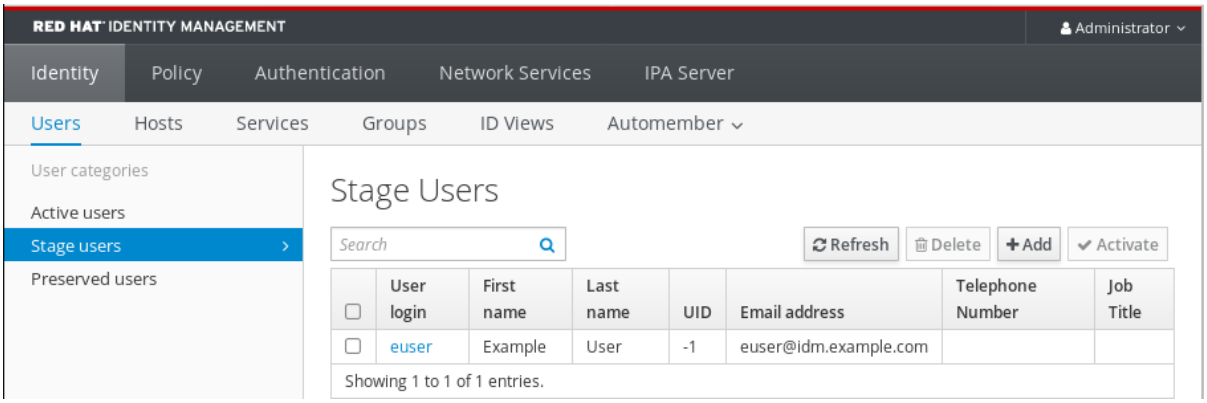
在用户可以登录到 IdM 之前，必须按照以下流程激活一个 stage 用户帐户，然后才能将用户添加到 IdM 组中。

**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。

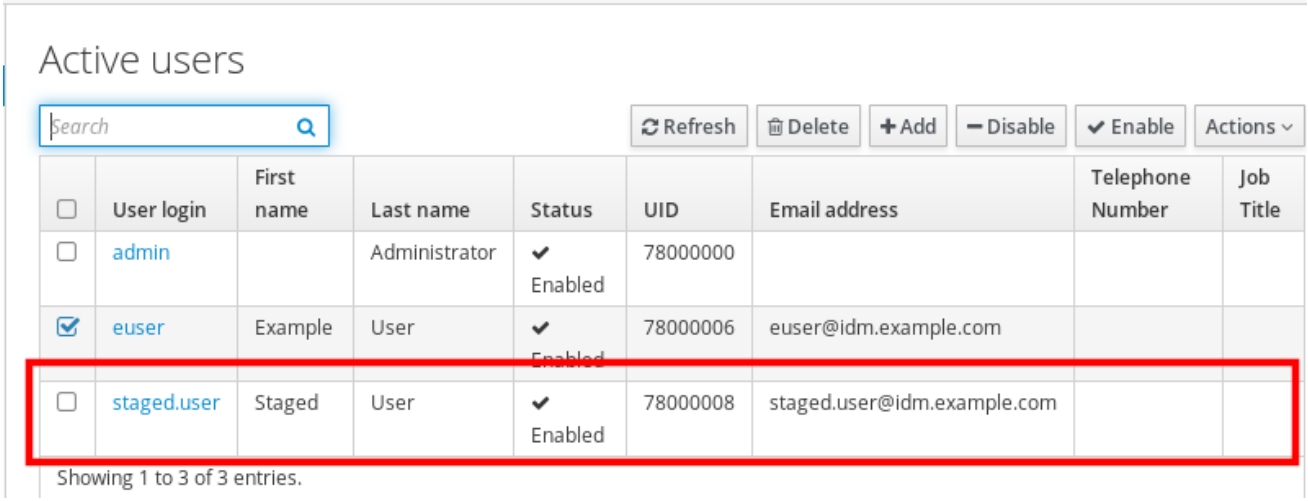
**步骤**

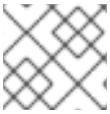
1. 登录到 IdM Web UI。
2. 进到 **Users → Stage users** 选项卡。
3. 单击您要激活的用户帐户的复选框。
4. 单击 **Activate** 按钮。




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。



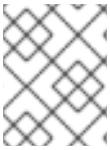
**注意**

在此阶段，您可以向用户组添加活动的用户帐户。

## 3.4. 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**

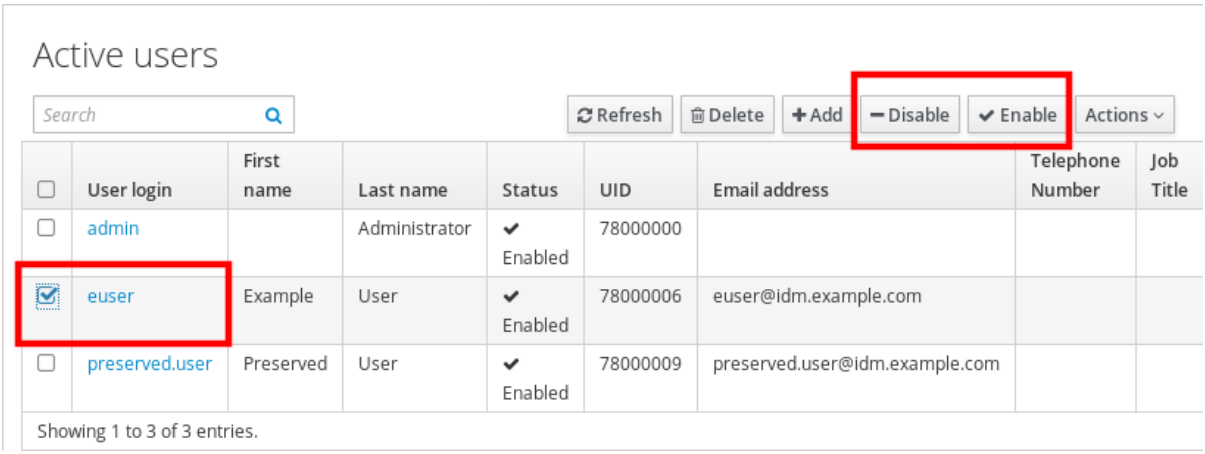
禁用用户帐户后，任何现有的连接都会保持有效，直到用户的 Kerberos TGT 和其他票据过期为止。票据过期后，用户将无法续订。

**先决条件**

- 管理 IdM Web UI 或用户管理员角色的管理员特权。

**步骤**

1. 登录到 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。

| User login | First name | Last name | Status | UID | Email address | Telephone Number |
|---|---|---|---|---|---|---|
| admin | | Administrator | Enabled | 78000000 | | |
| euser | Example | User | Disabled | 78000006 | euser@idm.example.com | |
| preserved.user | Preserved | User | Enabled | 78000009 | preserved.user@idm.example.com | |

## 3.5. 在 WEB UI 中启用用户帐户

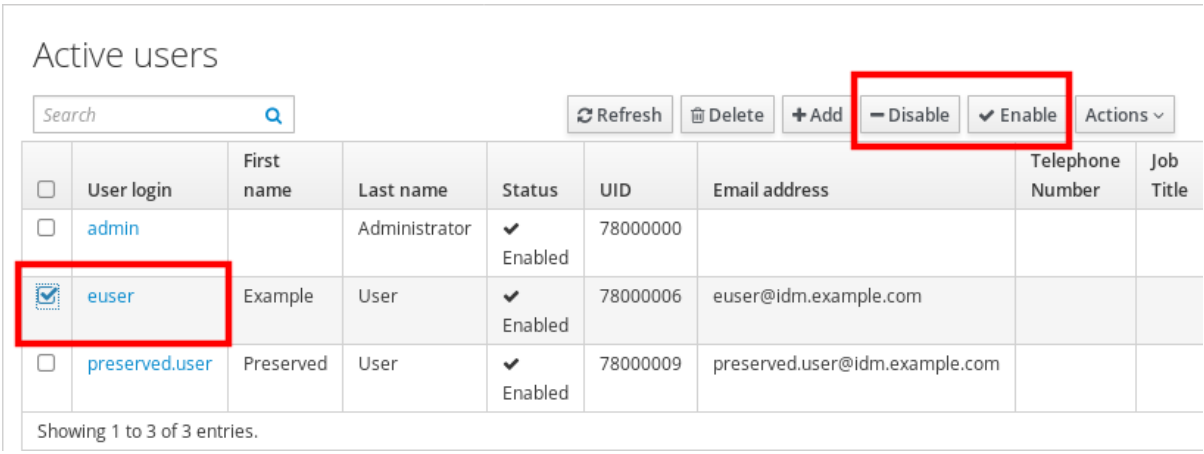
通过 IdM，您可以启用禁用的活动用户帐户。启用用户帐户可激活禁用的帐户。

**先决条件**

- 管理 IdM Web UI 或用户管理员角色的管理员特权。

**步骤**

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。

## 3.6. 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**

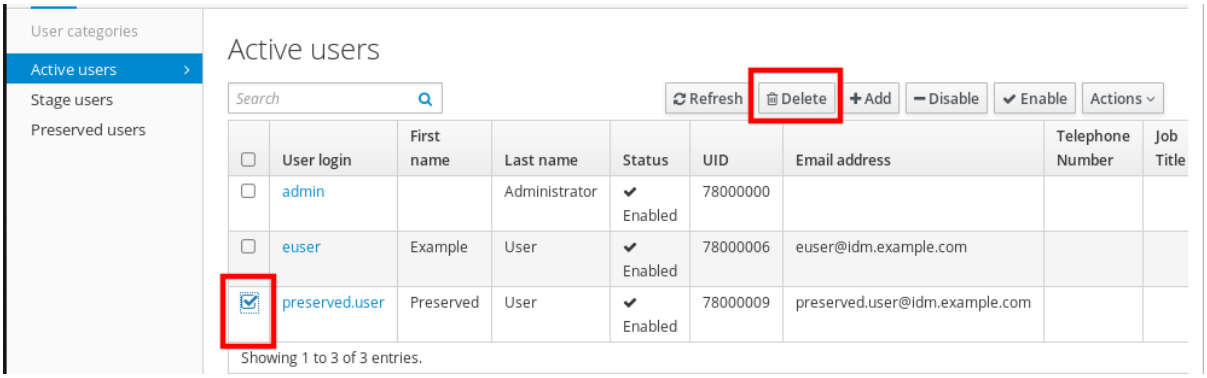
处于保留状态的用户列表可以提供过去用户帐户的历史记录。

**先决条件**

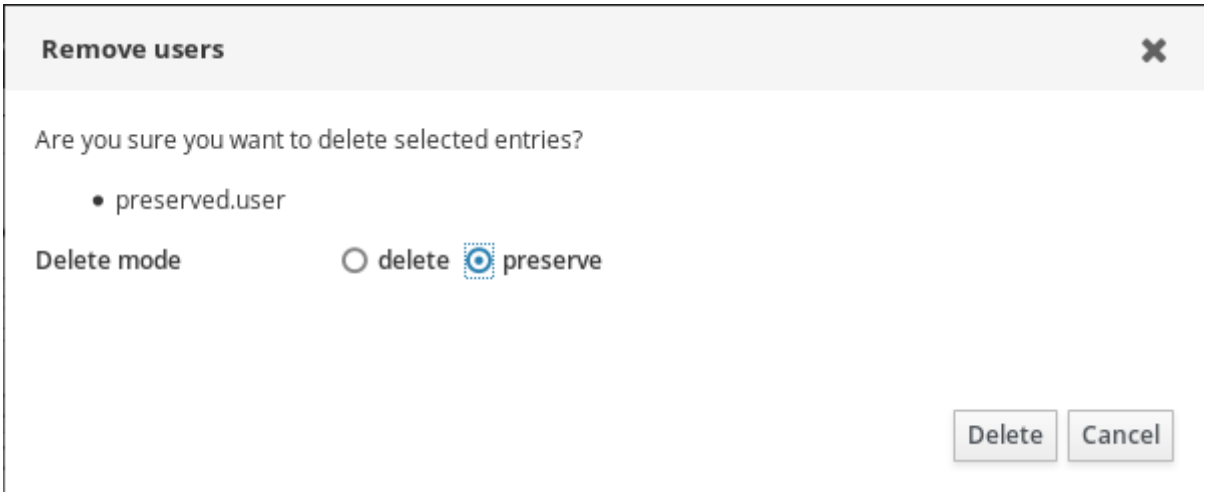
- 管理 IdM（身份管理）Web UI 或用户管理员角色的管理员特权。

**步骤**

1. 登录到 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 。

## 3.7. 在 IDM WEB UI 中恢复用户

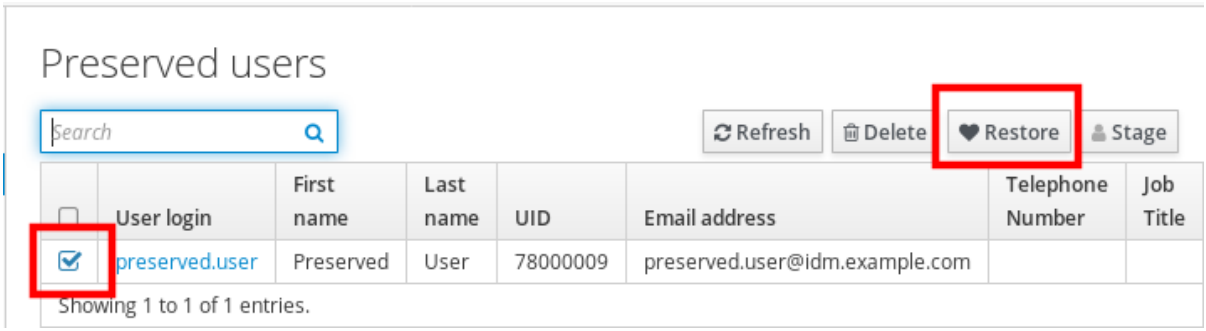
IdM （身份管理）可让您将保留的用户帐户恢复到活动状态。您可以将保留的用户恢复成活跃用户或 stage 用户。

**先决条件**

- 管理 IdM Web UI 或用户管理员角色的管理员特权。

**步骤**

1. 登录到 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。

## 3.8. 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。

**步骤**

1. 登录到 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。

用户帐户已从 IdM 永久删除。

# 第 4 章 使用 ANSIBLE PLAYBOOK 管理用户帐户

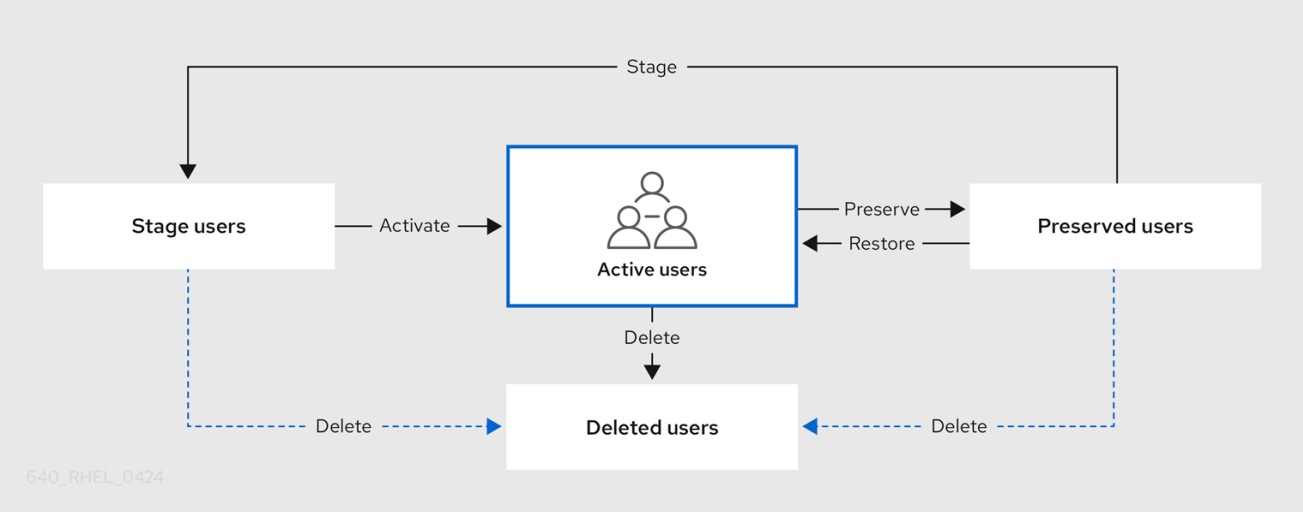
您可以使用 Ansible playbook 管理 IdM 中的用户。在介绍了用户生命周期后，本章将介绍如何将 Ansible playbook 用于以下操作：

- 确保存在一个单独的用户，这个用户直接列在 **YML** 文件中。
- 确保存在多个用户，这些用户直接列在 **YML** 文件中。
- 确保存在多个用户，这些用户在由 **YML** 文件引用的 **JSON** 文件中列出。
- 确保存在用户，用户直接在 **YML** 文件中列出。

## 4.1. 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。

## 4.2. 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，您已创建了一个带有 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 **ipaadmin_password**。
- 目标节点,也就是在其上执行 **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。

**步骤**

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：

```
---
```

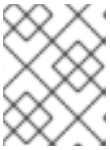
```
- name: Playbook to handle users
  hosts: ipaserver

  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。

## 4.3. 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，您已创建了一个带有 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 **ipaadmin_password**。
- 目标节点,也就是在其上执行 **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。

**步骤**

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

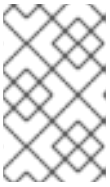
```
[ipaserver]
server.idm.example.com
```

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。

## 4.4. 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，您已创建了一个带有 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 **ipaadmin_password**。
- 目标节点,也就是在其上执行 **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。

**步骤**

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.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。

## 4.5. 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，您已创建了一个带有 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 **ipaadmin_password**。
- 目标节点,也就是在其上执行 **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。

**步骤**

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.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。

## 4.6. 其他资源

- 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。

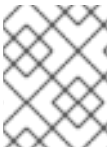
# 第 5 章 在 IDM 中管理用户密码

## 5.1. 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。

## 5.2. 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。

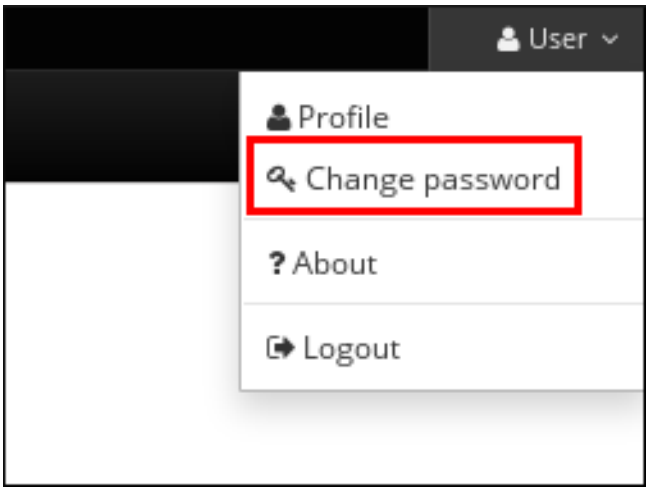
**先决条件**

- 已登陆到 IdM Web UI。

**流程**

1. 在右上角，点击 **User name → Change password**。

   **图 5.1. 重置密码**




2. 输入当前的密码以及新密码。

## 5.3. 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**

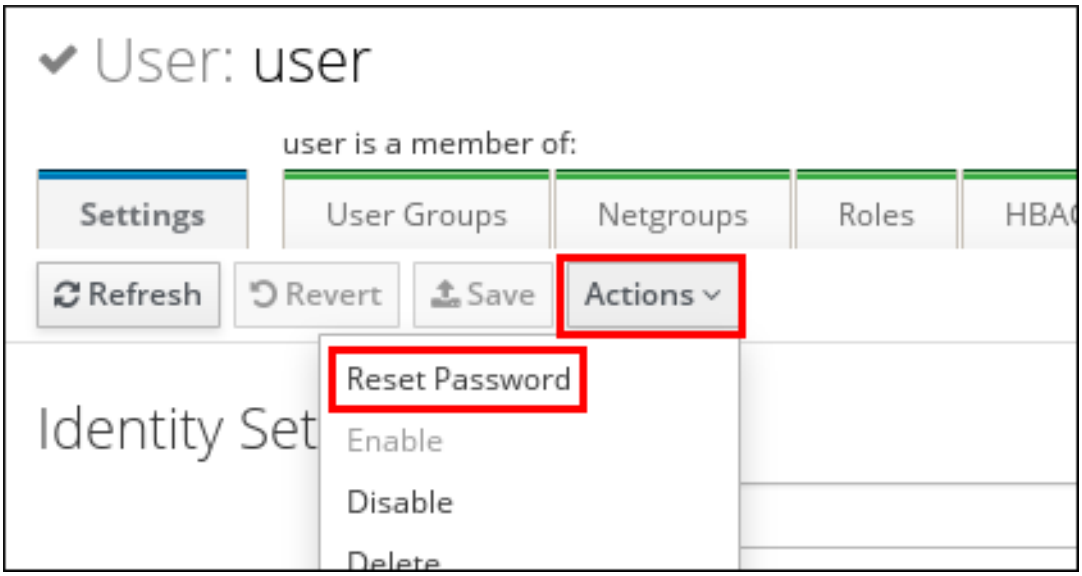
**先决条件**

- 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 5.2. 重置密码**




4. 输入新密码，然后单击 **Reset Password**。

**图 5.3. 确认新密码**


## 5.4. 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}AAAgABU0bKhyjY53NcxY33ueoPjOUWtl4iyYN5uW...
```

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}AAAgABU0bKhyjY53NcxY33ueoPjOUWtl4iyYN5uW...
```

   c. 启动服务器上安装的所有 IdM 服务：

```
# ipactl start
```

## 5.5. 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**。

## 5.6. 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。

## 5.7. 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

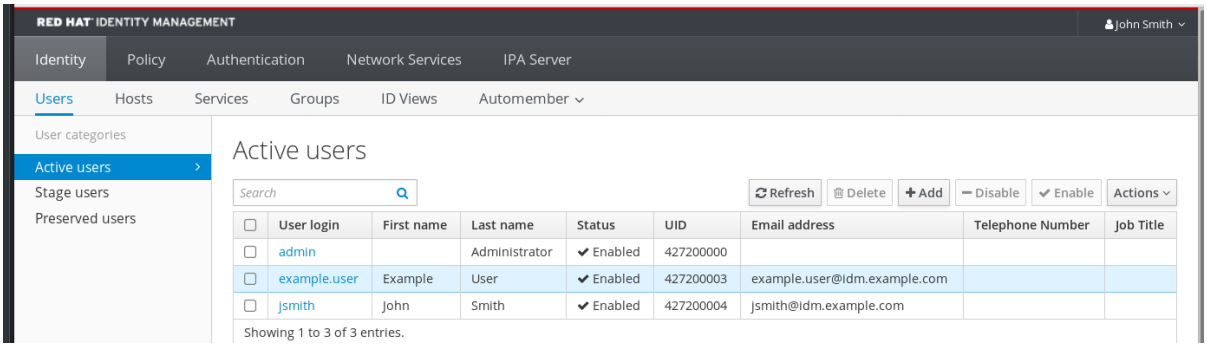
1. 显示用户帐户的状态，来查看失败的登录次数：

```
$ 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。

**其他资源**

- 在 IdM 中密码失败后解锁用户帐户

## 5.8. 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。

**其他资源**

- 检查 IdM 用户帐户是否已被锁住

## 5.9. 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 6 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。

## 6.1. 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。

## 6.2. 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 6.1. 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，如果属性设为 0，则密码永远不会过期。 | max lifetime = 180<br><br>用户密码仅 180 天有效。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 -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**

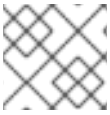
如果您一组不同的硬件可能不能使用国际字符和符号，则字符类要求应为英语字母和常用符号。有关密码中字符类策略的更多信息，请参阅红帽知识库中的密码中哪些字符有效？

## 6.3. 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，您已创建了一个带有 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 **ipaadmin_password**。
- 目标节点,也就是在其上执行 **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。
- 您知道 IdM 管理员密码。
- 正在确保 IdM 中存在密码策略的组。

**步骤**

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。

**其他资源**

- 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。

## 6.4. 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、RHEL 8 和 RHEL 9 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，IdM 客户端与在 RHEL 8.3 或更早版本中运行的 IdM 服务器进行通信，则系统管理员设置的新密码策略要求不会被应用。为确保一致性的行为，将所有服务器升级或更新到 RHEL 8.4 及更新的版本。

**其他资源：**

- 将额外密码策略应用到 IdM 组
- **pwquality(3)** man page

## 6.5. 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。

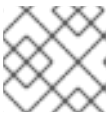
**先决条件**

- 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。

**步骤**

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 2 个以上连续相同的字符的密码。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。

**其他资源**

- IdM 中的额外密码策略

## 6.6. 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，来为特定的 IdM 组增强密码策略要求。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对应的用户名。
- 密码不包含连续两个相同的字符。
- 密码中的任何单调字符序列都不超过 3 个字符。这意味着系统不接受如 **1234** 或 **abcd**）这样序列的密码。

**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/* 目录中创建了一个带有 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。

**步骤**

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。

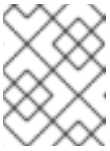
4. 在命令行中，尝试为 **test_user** 获取 Kerberos 票据授予票据 (TGT)：

```
$ 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。输入包含超过 3 个字符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. 验证是否您已获得 TGT，这只有在输入有效密码后才能获得：

```
$ 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
```

**其他资源**

- 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 7 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

## 7.1. 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）。

## 7.2. 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。

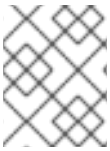
**步骤**

- 安装 EPN 工具：

```
# dnf install ipa-client-epn
```

## 7.3. 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，来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。

**步骤**

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。

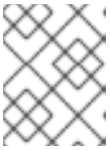
```
mail_from =admin-email@example.com
```

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**

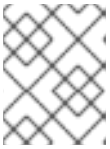
如果返回的用户列表非常大，并且运行工具时没有 **--dry-run** 选项，这可能会导致您的电子邮件服务器出现问题。

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。

**其他资源**

- 请参阅 **ipa-epn** 手册页。
- 请参阅 **epn.conf** 手册页。

## 7.4. 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，来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

**步骤**

- 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。

**其他资源**

- 请参阅 **ipa-epn** 手册页。

## 7.5. 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。

**步骤**

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。

**其他资源**

- 请参阅 **ipa-epn** 手册页。

# 第 8 章 为 IDM 客户端上的 IDM 用户授予 SUDO 访问权限

了解有关授予 **sudo** 访问身份管理中用户的更多信息。

## 8.1. 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 **集中的 sudo 规则**。

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。

**其他资源**

- 请参阅管理 sudo 权限。

## 8.2. 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。

**步骤**

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：

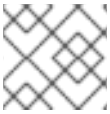
   a. 要定义 **sudo** 规则开始有效的时间，请使用带有 **--setattr sudonotbefore=*DATE*** 选项的 **ipa sudorule-mod *sudo_rule_name*** 命令。*DATE* 值必须遵循 **yyyymmddHHMMSSZ** 格式，以

秒为单位。例如，要将 **idm_user_reboot** 规则的有效性的开始时间设置为 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 规则不再有效的时间，请使用 **--setattr sudonotafter=DATE** 选项。例如：要将 **idm_user_reboot** 规则有效期结束的时间设置为 2026 年 12 月 31 日 12:34:00 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:
```

## 8.3. 使用 CLI 向 IDM 客户端上的 AD 用户授予 SUDO 访问权限

身份管理 (IdM) 系统管理员可以使用 IdM 用户组来设置 IdM 用户的访问权限、基于主机的访问控制、**sudo** 规则和其他控制。IdM 用户组授予和限制 IdM 域资源的访问权限。

您可以将 Active Directory (AD) *用户*和 AD *组*添加到 IdM 用户组。要做到这一点：

1. 将 AD 用户或组添加到 *non-POSIX* 外部 IdM 组中。

2. 将 non-POSIX 外部 IdM 组添加到 IdM *POSIX* 组。

然后，您可以通过管理 POSIX 组的权限来管理 AD 用户的特权。例如，您可以为特定命令授予特定 IdM 主机上的 IdM POSIX 用户组的 **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。这样，通过在单个 AD 域中保留用户和组管理，可以更轻松地为 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，来为 **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，这通常为 **root** 用户保留。**administrator@ad-domain.com** 是 **ad_users_external** non-POSIX 组的成员，后者又是 **ad_users** POSIX 组的成员。

**先决条件**

- 您已获得 IdM **admin** Kerberos 票据授予票 (TGT)。
- IdM 域和 **ad-domain.com** AD 域之间存在跨林信任。
- **idmclient** 主机上没有本地的 **administrator** 帐户 ：**administrator** 用户没有列在本地 **/etc/passwd** 文件中。

**流程**

1. 创建 ***ad_users*** 组，它包括带有 **administrator@ad-domain** 成员的 ***ad_users_external*** 组：

   a. *可选* ：创建或选择 AD 域中的对应组，用来管理 IdM 域中的 AD 用户。您可以使用多个 AD 组，并将它们添加到 IdM 端的不同组中。

   b. 创建 ***ad_users_external*** 组，并通过添加 **--external** 选项来指示它包含 IdM 域外部的成员：

```
[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带有 **global** 或 **universal** 组范围的 AD 安全组，如 Active Directory 安全组文档中所述。例如，**Domain users** 或 **Domain admins** AD 安全组不能使用，因为组的范围是 **domain local**。

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 用户必须通过完全限定名称来标识，如 **DOMAIN\user_name** 或 **user_name@DOMAIN**。AD ID 然后会被映射到用户的 AD SID。这同样适用于添加 AD 组。

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. 授予 **ad_users** 成员在 **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，这是 **ad_users** 组的间接成员：

```
$ ssh administrator@ad-domain.com@ipaclient
Password:
```

2. 另外，还可显示 **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** 密码：

```
[administrator@ad-domain.com@idmclient ~]$ sudo /usr/sbin/reboot
[sudo] password for administrator@ad-domain.com:
```

**其他资源**

- 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

## 8.4. 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。

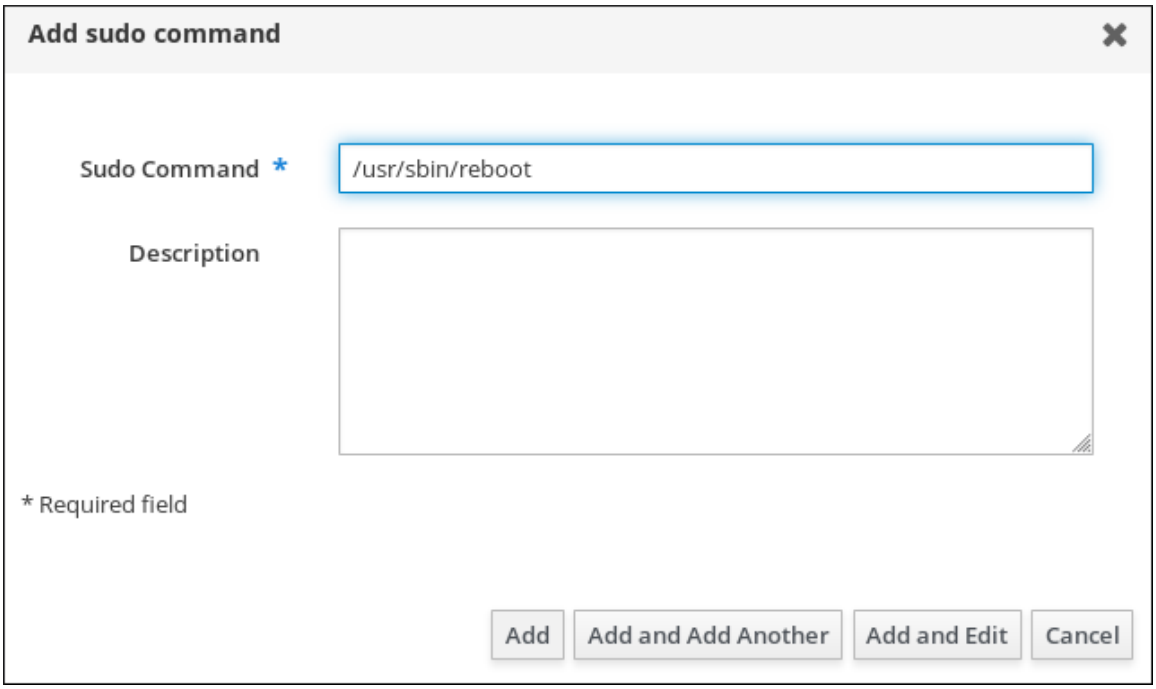
**先决条件**

- 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。

**步骤**

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 8.1. 添加 IdM sudo 命令




d. 点 **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：

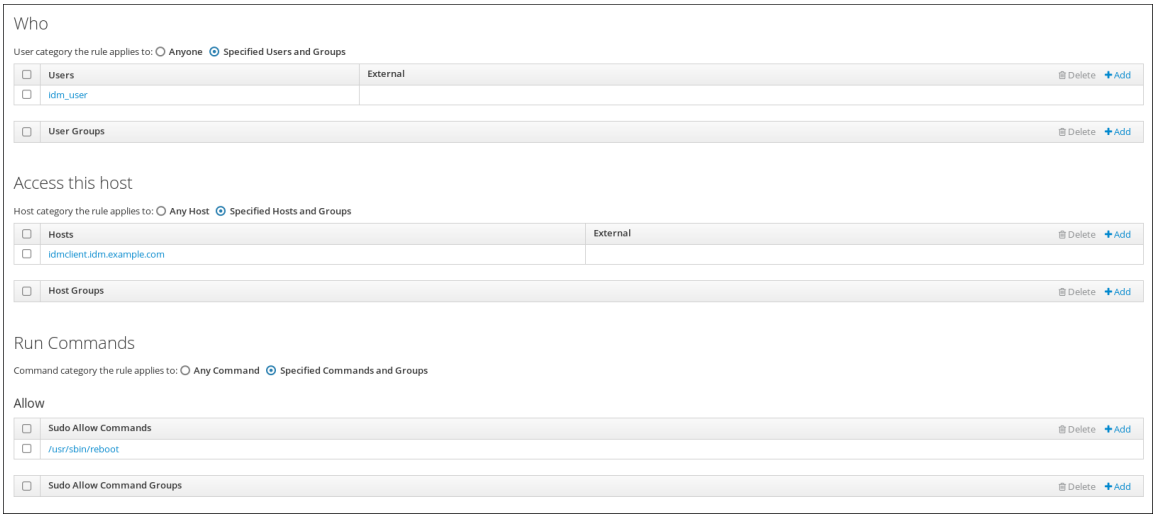
i. 在 **Run Commands** 一节的 **Command category the rule applies to**子小节中，选择 **Specified Commands and Groups**单选按钮。

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 8.2. 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。

## 8.5. 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。

使用这个示例在命令行上创建一个名为 **run_third-parent-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。
- 您已在 **idmclient** 主机上已安装了一个名为 **third-party-app** 的自定义应用程序。
- 用于 **third-party-app** 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 **thirdrdapp** 的本地服务帐户，来执行 **third-party-app** 应用程序的命令。

**步骤**

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 数据库中：

```
[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（或用 **--groups=*** 选项指定的组）可以是来自 IdM 外部，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.
```

## 8.6. 在 IDM WEBUI 中创建一个 SUDO 规则，该规则以 IDM 客户端上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。

使用这个示例来在 IdM WebUI 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。
- 您已在 **idmclient** 主机上已安装了一个名为 **third-party-app** 的自定义应用程序。
- 用于 **third-party-app** 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 **thirdrdapp** 的本地服务帐户，来执行 **third-party-app** 应用程序的命令。

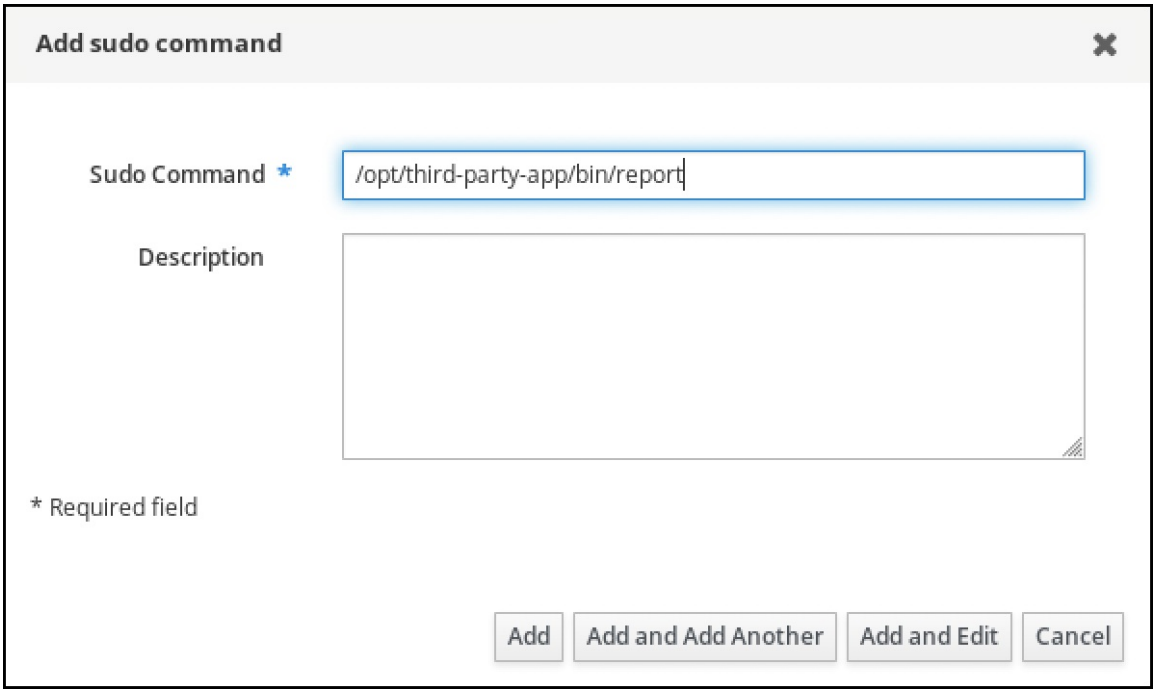
**步骤**

1. 将 **/opt/third-party-app/bin/report** 命令添加到 **sudo** 命令的 IdM 数据库中：

   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**。



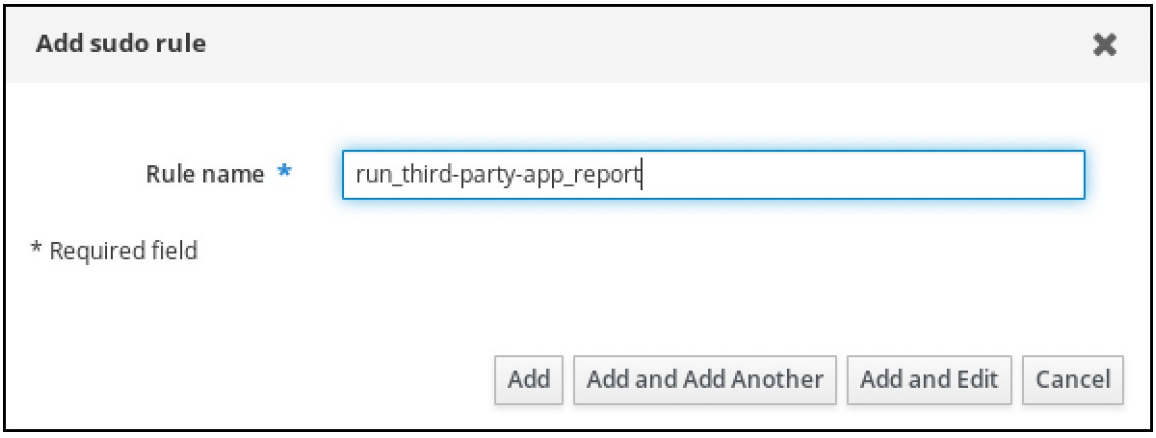

d. 点 **Add**。

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**。



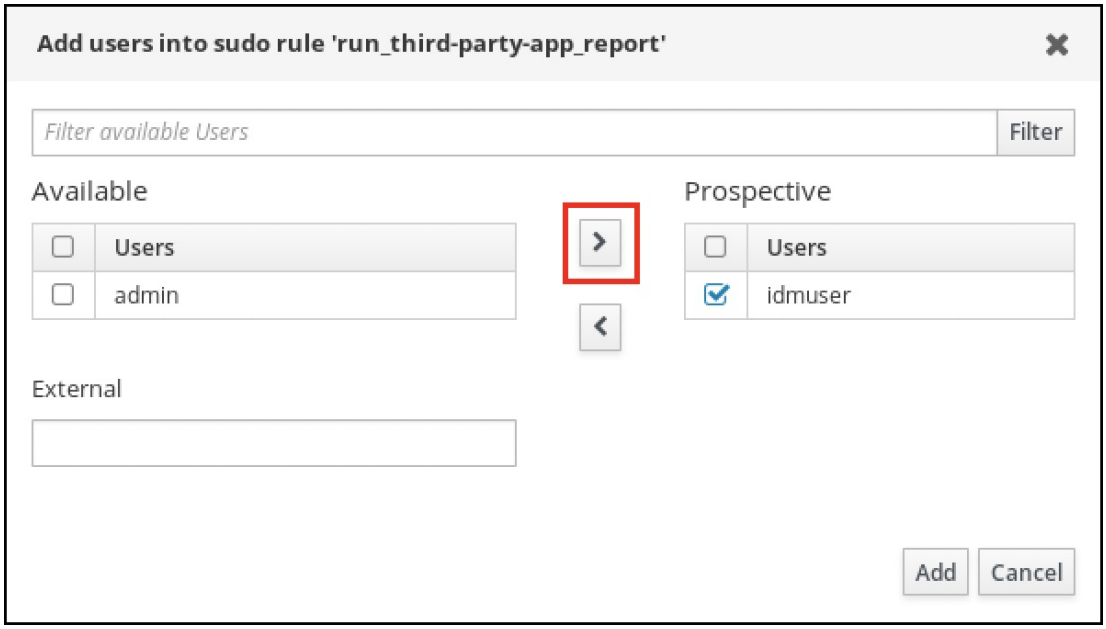

d. 点 **Add and Edit**。

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 **User category the rule applies to**子部分中，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。



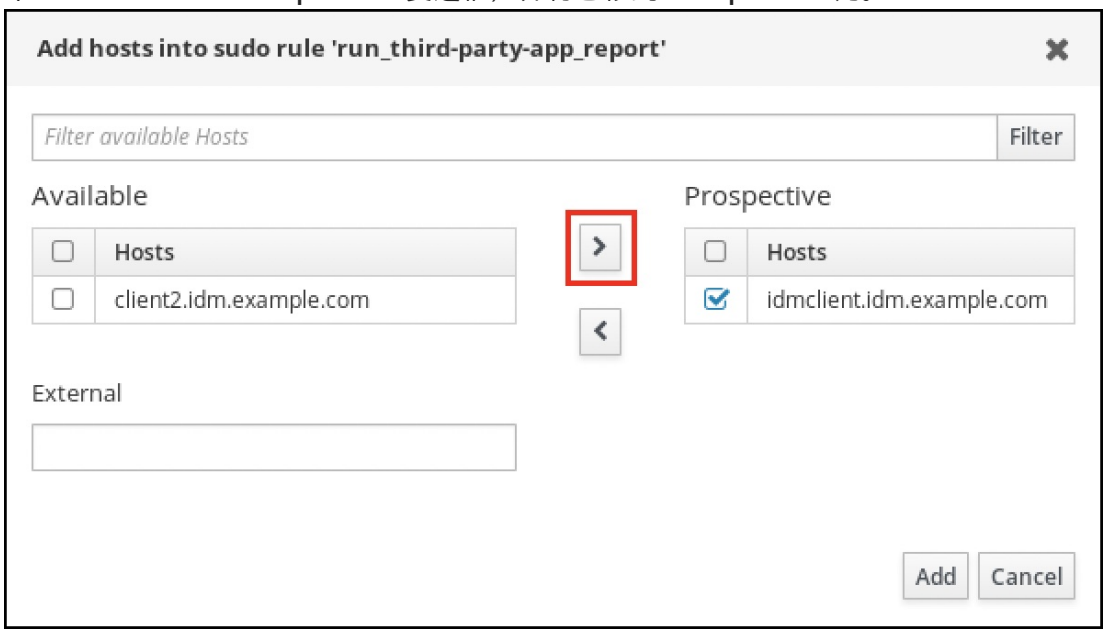

iv. 点 **Add**。

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** 子部分中，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 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。



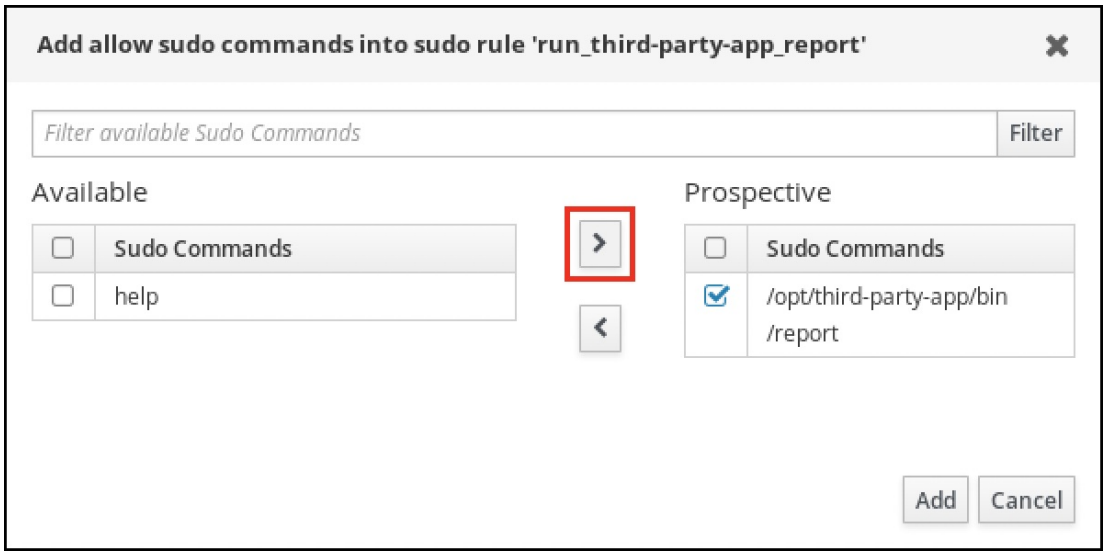

iv. 点 **Add**。

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** 子部分中，单击 **Add** 来打开 **将允许 sudo 命令添加到 sudo 规则 "run_third-party-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。



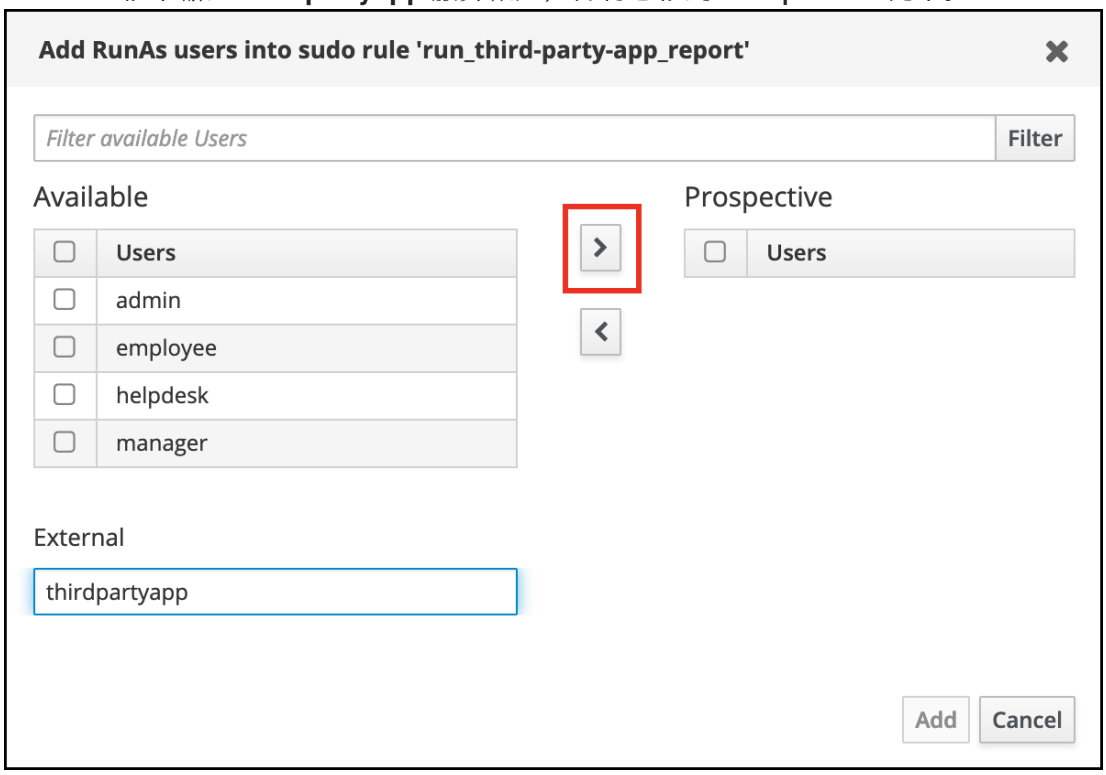

iv. 单击 **Add** 以返回到 **run_third-party-app_report** 页。

h. 指定 RunAs 用户：

i. 在 **As Whom** 部分中，选中 **Specified Users and Groups** 单选按钮。

ii. 在 **RunAs Users** 子部分中，单击 **Add** 以将 **Add RunAs** 用户打开 sudo 规则 "run_third-party-app_report" 对话框。

iii. 在 **Add RunAs users in sudo rule "run_third-party-app_report"** 对话框中，在 **External** 框中输入 **thirdpartyapp** 服务帐户，并将它移到 **Prospective** 列中。

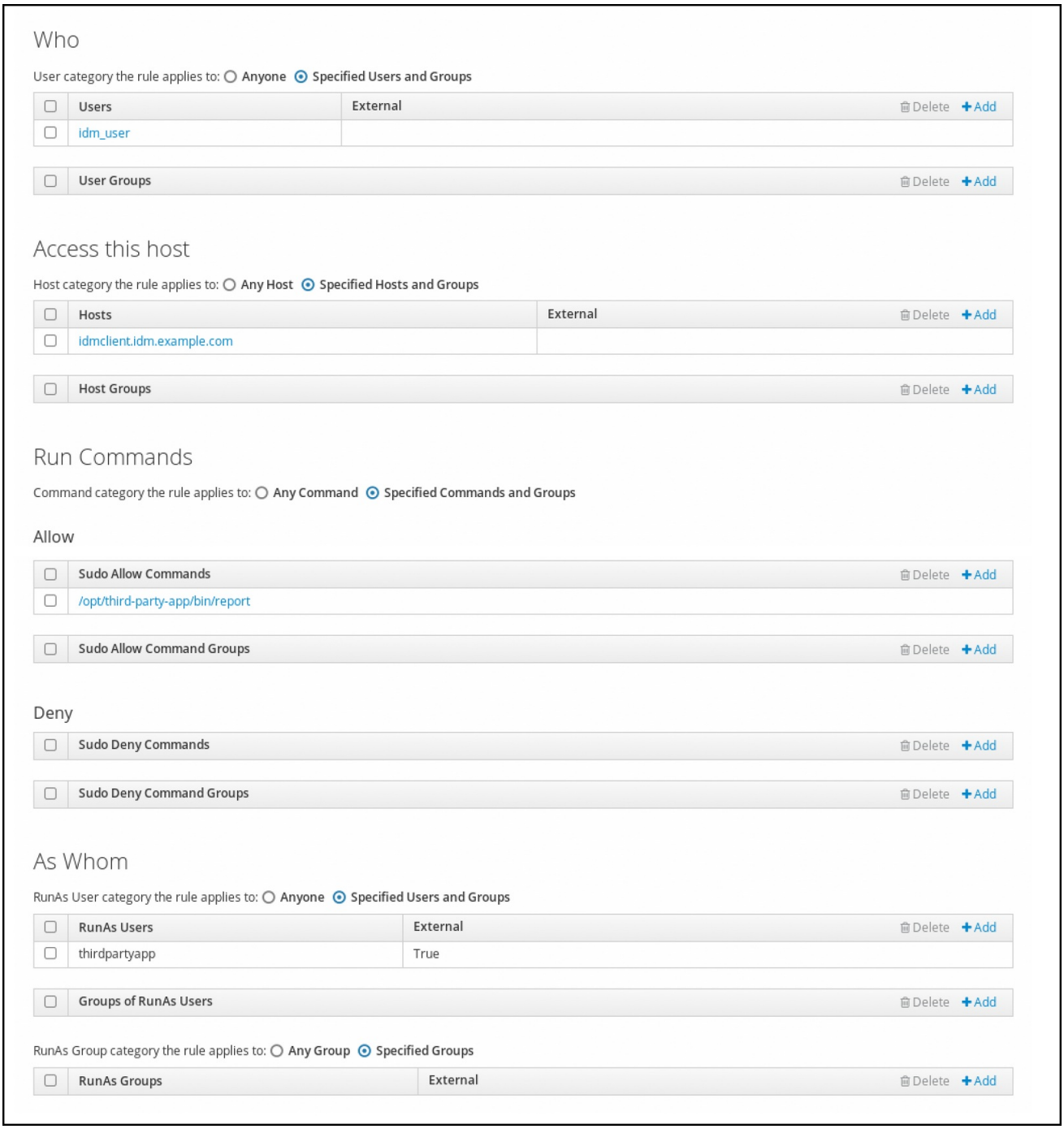



iv. 单击 **Add** 以返回到 **run_third-party-app_report** 页。

i. 单击左上角的 **Save**。

新规则默认为启用。

**图 8.3. sudo 规则的详细信息**




**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.
```

## 8.7. 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。有了这个配置，IdM 用户可以使用他们的 Kerberos ticket 向 **sudo** 命令进行身份验证。

**先决条件**

- 您已为 IdM 用户创建了一个应用于 IdM 主机的 **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。

**步骤**

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
```

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 9.2 或更高版本：

a. [可选] 确定您是否选择了 **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 9.1 或更早版本：

a. 打开 **/etc/pam.d/sudo** PAM 配置文件。

b. 添加下列条目，来作为 **/etc/pam.d/sudo** 文件中 **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** 帐户的身份登录主机。

```
[root@idm-client ~]# ssh -l idm_user@idm.example.com localhost
idm_user@idm.example.com's password:
```

2. 验证您作为 **idm_user** 帐户有一个票据授予票据。

```
[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** 而不指定密码来重新启动机器。

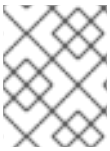
```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
```

**其他资源**

- IdM 术语 列表中的 GSSAPI 条目
- 使用 IdM Web UI 向 IdM 客户端上的 IdM 用户授予 sudo 访问权限
- 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

## 8.8. 在 IDM 客户端上启用 GSSAPI 身份验证，并为 SUDO 强制使用 KERBEROS 身份验证指示符

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，只有使用智能卡登录的用户才能使用 Kerberos 票据对这些命令进行身份验证。



**注意**

您可以将此流程用作模板，来其他 PAM 感知服务配置 带有 SSSD 的 GSSAPI 身份验证，并进一步限制对拥有附加到 Kerberos 票据的特定身份验证指示符的用户的访问。

**先决条件**

- 您已为 IdM 用户创建了一个应用于 IdM 主机的 **sudo** 规则。在本例中，您已创建了 **idm_user_reboot sudo** 规则，来为 **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。
- 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。

**步骤**

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，来作为 **/etc/pam.d/sudo** 文件中 **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，来作为 **/etc/pam.d/sudo-i** 文件中 **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** 帐户身份登录到主机，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. 验证您是否有一个智能卡用户的票据授予票据。

```
[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** 而不指定密码来重新启动机器。

```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
```

**其他资源**

- 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 向 IdM 客户端上的 IdM 用户授予 sudo 访问权限
- 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

## 8.9. SSSD 选项控制 PAM 服务的 GSSAPI 身份验证

您可以在 **/etc/sssd/sssd.conf** 配置文件中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以逗号分隔的 PAM 服务的列表，这些服务允许使用 **pam_sss_gss.gss.so** PAM 模块来尝试 GSSAPI 身份验证。要明确禁用 GSSAPI 身份验证，请将这个选项设为 **-**。

pam_gssapi_indicators_map

这个选项只适用于身份管理(IdM)域。使用此选项可以列出向服务授予 PAM 访问权限所需的Kerberos 身份验证指示符。对的格式必须是 ***<PAM_service>: _<required_authentication_indicator>_***。有效的验证指示符为：

- **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 模块将验证能够获取所需服务票据的每个用户。

**示例**

以下选项为 **sudo** 和 **sudo-i** 服务启用 Kerberos 身份验证，要求 **sudo** 用户通过一次性密码进行身份验证，并且用户名必须与 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，以覆盖 **[pam]** 部分中的任何全局值。以下选项将不同的 GSSAPI 设置应用到每个域：

**对于 idm.example.com 域**

- 为 **sudo** 和 **sudo -i** 服务启用 GSSAPI 身份验证。
- 需要 **sudo** 命令的验证证书或智能卡验证器。
- 需要 **sudo -i** 命令的一次性密码身份验证器.
- 强制实施匹配用户名和 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
...
```

**其他资源**

- Kerberos 认证指示符

## 8.10. 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，请使用以下场景来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/pam.d/** 目录中的 **/etc/sssd/sssd.conf** 文件和 PAM 文件。

**步骤**

- 如果您看到以下错误，则 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，在 **/etc/krb5.conf** Kerberos 配置文件中的 **[domain_realm]** 部分中直接添加主机名：

```
[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，或者使用 SSSD 进行验证：

```
[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，验证您是否使用 SSSD 进行了身份验证，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，您可以为 **pam_sss_gss.so** PAM 模块启用调试输出。
  - 在 PAM 文件中的所有 **pam_sss_gss.so** 条目的末尾添加 **debug** 选项，如 **/etc/pam.d/sudo** 和 **/etc/pam.d/sudo-i** ：

```
[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
```

## 8.11. 使用 ANSIBLE PLAYBOOK 来确保 IDM 客户端上 IDM 用户的 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，您已创建了一个带有 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 **ipaadmin_password**。

- 目标节点,也就是在其上执行 **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。
- 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。

**步骤**

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，以确保在 **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. 创建引用命令的 **sudo** 规则：

   a. 创建一个 **ensure-sudorule-for-idmuser-on-idmclient-is-present.yml** Ansible playbook，其使用 **sudo** 命令条目来确保存在 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** 重新启动 **idmclient**，来测试您在 IdM 服务器上已确认存在的 **sudo** 规则可以在 **idmclient** 上正常工作。请注意，服务器上所做的更改可能需要几分钟才能在客户端上生效。

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**，则机器将重新启动。

**其他资源**

- 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 9 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。您也可以在使用 LDAP 数据交换格式 (LDIF) 的文件中为 **ldapmodify** 命令提供数据。

## 9.1. 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
```

## 9.2. 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
```

## 9.3. 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 9.1. 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 9.2. 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
```

**其他资源**

- 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。

## 9.4. 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。

**步骤**

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 10 章 使用 LDAPSEARCH 命令搜索 IDM 条目

您可以使用 **ipa find** 命令通过 Identity Management 条目进行搜索。有关 **ipa** 命令的更多信息，请参阅 IPA 命令的结构 部分。

本节介绍了通过 Identity Management 条目使用 **ldapsearch** 命令行命令的替代搜索选项的基础知识。

## 10.1. 使用 LDAPSEARCH 命令

**ldapsearch** 命令具有以下格式：

```
# ldapsearch [-x | -Y mechanism] [options] [search_filter] [list_of_attributes]
```

- 要配置身份验证方法，请指定 **-x** 选项以使用简单绑定或 **-Y** 选项来设置简单验证和安全层 (SASL) 机制。请注意，如果您使用 **-Y GSSAPI** 选项，则需要获取 Kerberos ticket。
- *options* 是 **ldapsearch** 命令的选项，它包括在下表中。
- *search_filter* 是一个 LDAP 搜索过滤器。
- *list_of_attributes* 是搜索结果返回的属性列表。

例如，要为用户名 *user01* 搜索基本 LDAP 树的所有条目：

```
# ldapsearch -x -H ldap://ldap.example.com -s sub "(uid=user01)"
```

- **x** 选项告知 **ldapsearch** 命令通过简单绑定进行身份验证。请注意，如果您没有使用 **-D** 选项提供可辨识名称 (DN)，则身份验证是匿名的。
- **-H** 选项将您连接到 *ldap://ldap.example.com*。
- **-s sub** 选项告知 **ldapsearch** 命令从基本 DN 开始搜索所有名为 *user01* 的用户。*"(uid=user01)"* 是一个过滤器。

请注意，如果没有通过 **-b** 选项提供搜索的起点，则命令会在默认树中搜索。它在 **etc/openldap/ldap.conf** 文件的 BASE 参数中指定。

**表 10.1. ldapsearch 命令选项**

| 选项 | 描述 |
|---|---|
| -b | 搜索的起点。如果您的搜索参数包含星号 (*) 或其他字符，命令行可以解释为代码，则必须以单引号或双引号括起该值。例如，**-b cn=user,ou=Product Development,dc=example,dc=com**。 |
| -D | 要进行身份验证的可辨识名称 (DN)。 |
| -H | 连接到服务器的 LDAP URL。**-H** 选项替换了 **-h** 和 **-p** 选项。 |
| -l | 等待搜索请求完成的时间限制（以秒为单位）。 |

| 选项 | 描述 |
| --- | --- |
| -s *scope* | 搜索的范围。对于范围，您可以选择以下之一：<br>• **base** 仅搜索来自 **-b** 选项，或者由 **LDAP_BASEDN** 环境变量定义的条目。<br>• **one** 仅搜索来自 **-b** 选项的条目的子条目。<br>• **sub** 以 -b 选项作为开始点进行的子树搜索。 |
| -W | 对密码的请求。 |
| -x | 禁用默认 SASL 连接以允许简单的绑定。 |
| -Y *SASL_mechanism* | 为身份验证设置 SASL 机制。 |
| -z *number* | 搜索结果中的最大条目数。 |

请注意，您必须使用 **ldapsearch** 命令通过 **-x** 或 **-Y** 选项指定一个验证机制。

**其他资源**

- 有关如何使用 **ldapsearch** 的详情，请参考 **ldapsearch (1)** man page。

## 10.2. 使用 LDAPSEARCH 过滤器

**ldapsearch** 过滤器允许您缩小搜索结果范围。

例如，您希望搜索结果包含将通用名称设置为 *example* 的所有条目：

```
"(cn=example)"
```

在本例中，*等号(=)* 是操作符，*example* 是值。

**表 10.2. ldapsearch 过滤器操作符**

| 搜索类型 | 操作符 | 描述 |
| --- | --- | --- |
| 相等 | = | 返回与值完全匹配的条目。例如：*cn=example*。 |
| 子字符串 | *=string* string* | 返回所有带有子字符串匹配的条目。例如，*cn=exa*l*。星号 (*) 表示零 (0) 或多个字符。 |
| 大于或等于 | >= | 返回所有带有大于或等于值的属性的条目。例如，*uidNumber >= 5000*。 |

| 搜索类型 | 操作符 | 描述 |
| --- | --- | --- |
| 小于或等于 | <= | 返回所有其属性小于或等于值的条目。例如，*uidNumber <= 5000*。 |
| 存在 | =* | 返回含有一个或多个属性的所有条目。例如：*cn=**。 |
| 大约 | ~= | 返回与值属性类似的所有选项。例如，*l~=san fransico* 可以返回 *l=san francisco*。 |

您可以使用 *boolean* 运算符将多个过滤器组合到 **ldapsearch** 命令中。

**表 10.3. ldapsearch 过滤器布尔值操作符**

| 搜索类型 | 操作符 | 描述 |
| --- | --- | --- |
| 和 | & | 返回过滤器中的所有语句都为 true 的所有条目。例如，*(&(filter)(filter)(filter)...)*。 |
| 或 | \| | 返回过滤器中至少有一个语句为 true 的所有条目。例如，*(\|(filter)(filter)(filter)...)*。 |
| 非 | ! | 返回过滤器中声明不为 true 的所有条目。例如，*(!(filter))*. |

# 第 11 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 的 CLI或 使用 LDIF 文件添加单个 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

## 11.1. 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。

**步骤**

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。

## 11.2. 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。

**步骤**

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.
```

## 11.3. 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**

**步骤**

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"
```

## 11.4. 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**

**步骤**

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。

## 11.5. 其他资源

- 请参阅 使用 ldapmodify 在外部管理 IdM 用户 。

# 第 12 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。

## 12.1. 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:
```

## 12.2. 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
```

## 12.3. 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:
```

## 12.4. 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 13 章 使用 PAC 信息增强 KERBEROS 安全性

从 RHEL 8.5 开始，您默认可以使用带有 Privilege Attribute 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)。

## 13.1. IDM 中使用特权属性证书 (PAC)

为提高安全性，RHEL Identity Management (IdM) 现在在新部署中默认发出带有 Privilege Attribute 证书 (PAC) 信息的 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。

默认情况下，Microsoft Active Directory (AD)使用的 SID 是从不重复使用的全局唯一标识符。SID 表达多个命名空间：每个域都有一个 SID，它是每个对象的 SID 中的前缀。

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生成 SID。这允许 IdM 使用 PAC 数据。如果您在 RHEL 8.5 之前安装了 IdM，且您尚未配置 AD 域的信任，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)。

通过在 Kerberos 票据中评估 PAC 信息，您可以使用更详细的信息来控制资源访问。例如，一个域中的 Administrator 帐户的 SID 与任何其他域中的 Administrator 帐户不同。在对 AD 域的带有信任的 IdM 环境中，您可以根据全局唯一的 SID 设置访问控制，而不是在不同位置中重复的简单用户名或 UID，如每个 Linux **root** 帐户都有 UID 0。

## 13.2. 在 IDM 中启用安全标识符 (SID)

如果您在 RHEL 8.5 之前安装了 IdM，且您还没有配置 AD 域的信任，您可能没有为 IdM 对象生成安全标识符(SID)。这是因为之前，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，这对基于 Privilege Access 证书 (PAC) 信息的安全性是必需的。

**先决条件**

- 在 RHEL 8.5 之前已安装了 IdM。
- 还没有运行 **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，以便为现有的用户和组生成 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，其具有以 **-500** 结尾的 SID，为域管理员保留 SID：

```
[root@server ~]# ipa user-show admin --all | grep ipantsecurityidentifier
  ipantsecurityidentifier: S-1-5-21-2633809701-976279387-419745629-500
```

-

**其他资源**

- IdM 中使用特权属性证书 (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 配置集成到基础 IPA 安装程序中

# 第 14 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

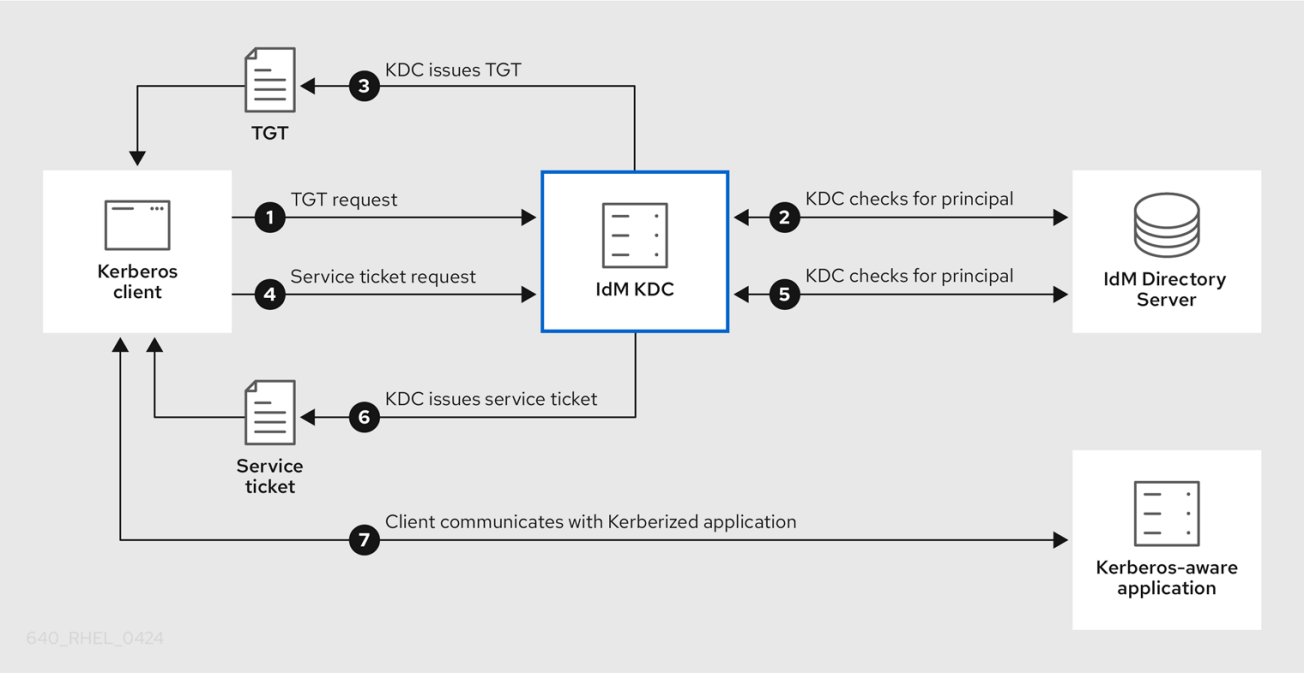
## 14.1. IDM KDC 的角色

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。

## 14.2. 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。

## 14.3. 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]

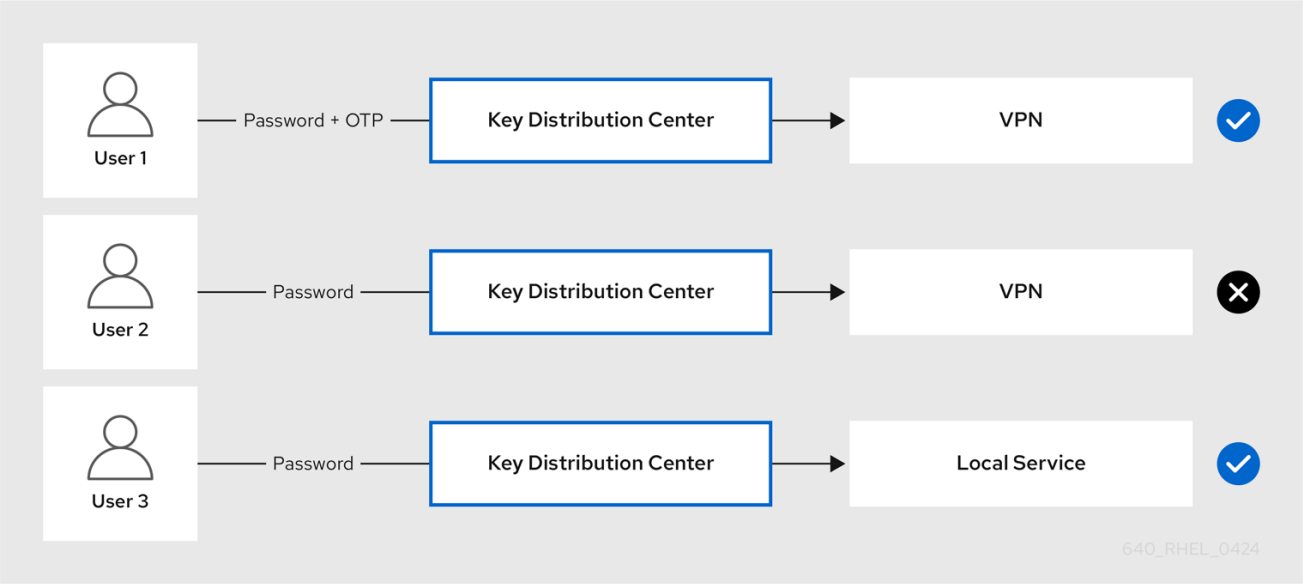
然后 KDC 将来自 TGT 的身份验证指标附加到来自它的任何服务票据请求。KDC 强制执行基于验证指标的策略，如服务访问控制、最长票据生命周期和最长续订期限。

### 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，要要求双因素身份验证连接到虚拟专用网络(VPN)，请将 **otp** 身份验证指标与该服务相关联。只有使用一次性密码从 KDC 获取初始 TGT 的用户才能登录到 VPN：

图 14.1. 需要 otp 验证指示符的 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。

**其他资源**

- 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

## 14.4. 为 IDM 服务强制执行身份验证指标

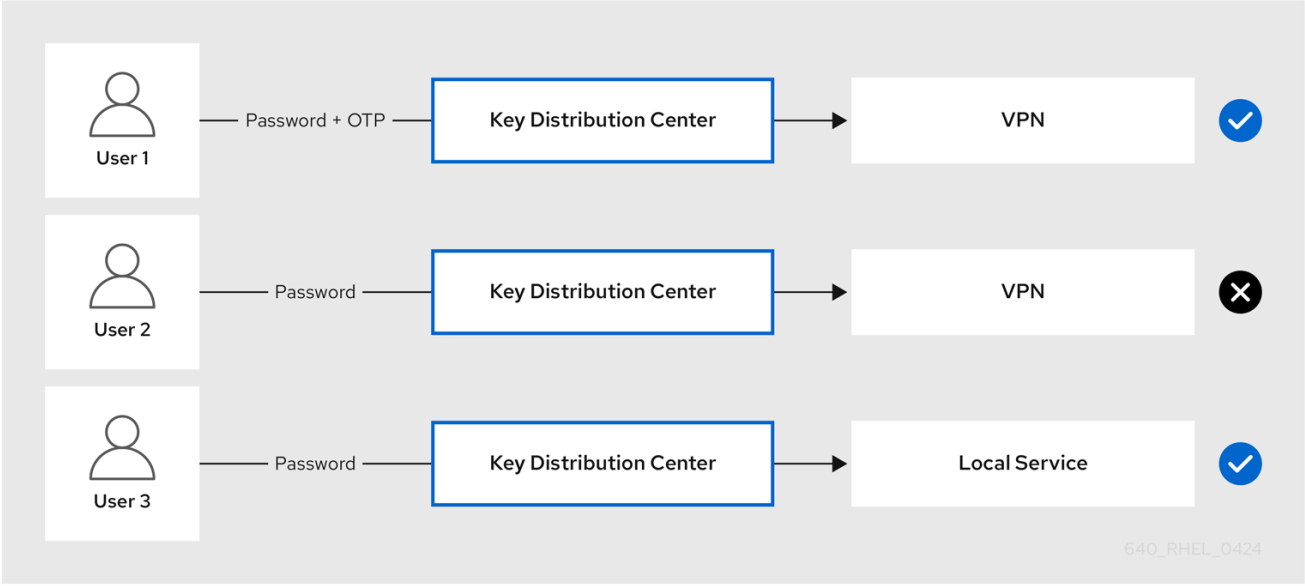
身份验证(IdM)支持的验证机制在身份验证强度方面存在差异。例如，使用一次性密码(OTP)与标准密码(OTP)的结合来获取初始 Kerberos 票据授予票(TGT)被视为比仅使用标准密码进行身份验证更加安全。

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 14.2. 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。

### 14.4.1. 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。

**步骤**

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)
```

### 14.4.2. 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。

按照以下流程将服务配置为需要来自传入服务票据请求的特定的 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
```

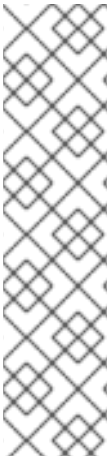
**步骤**

- 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
```

**其他资源**

- 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

### 14.4.3. 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。

**步骤**

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**。

**其他资源**

- 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

### 14.4.4. 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)。

**步骤**

- 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
```

### 14.4.5. 其他资源

- 请参阅 Kerberos 身份验证指标。

## 14.5. 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）

**步骤**

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
```

**其他资源**

- 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。

## 14.6. 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。

**步骤**

- 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。

**其他资源**

- 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。

## 14.7. 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）

**步骤**

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
```

**其他资源**

- 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。

## 14.8. 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 。

**步骤**

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
```

**其他资源**

- 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。

## 14.9. 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 14.1. 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** |

| 身份验证指标 | 最长生命周期的参数 | 最长续订期限的参数 |
| --- | --- | --- |
| **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 15 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。

## 15.1. 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 15.1. 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。 |

## 15.2. 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...]
```

## 15.3. 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，如果您使用 **ipa-server-install** 或 **ipa-replica-install** 工具传递了 **--no-pkinit** 选项，则服务器会在禁用 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，您可以使用命令来查找启用了 PKINIT 的所有服务器。

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** 命令检查是否已添加了 CA 证书。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*** 颁发。
- 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 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. 如果您使用带有 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，该命令请求来自 CA 的 PKINIT KDC 证书。

**其他资源**

- **ipa-server-certinstall(1)** 手册页

## 15.4. 其他资源

- 有关 Kerberos PKINIT 的详情，请参阅 MIT Kerberos 文档中的 PKINIT 配置。

# 第 16 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，以允许服务使用 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 文件内容列表
- 查看 IdM 主密钥的加密类型。

## 16.1. IDENTITY MANAGEMENT 如何使用 KERBEROS KEYTAB 文件

Kerberos keytab 是包含 Kerberos 主体及其对应加密密钥的文件。主机、服务、用户和脚本可以使用 keytab 安全地对 Kerberos 密钥分发中心 (KDC) 进行身份验证，而无需人工交互。

IdM 服务器上的每个 IdM 服务都有存储在 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 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，默认的 keytab 文件 **/etc/krb5.keytab** 存储 **host** 主体，这表示计算机在 Kerberos 域中，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)
```

**其他资源**

- 验证 Kerberos keytab 文件是否与 IdM 数据库同步

- IdM Kerberos keytab 文件内容列表

## 16.2. 验证 KERBEROS KEYTAB 文件是否与 IDM 数据库同步

当您更改 Kerberos 密码时，IdM 会自动生成一个新的对应的 Kerberos 密钥，并递增其密钥版本号 (KVNO)。如果没有使用新密钥和 KVNO 更新 Kerberos keytab，则任何依赖于该 keytab 来检索有效密钥的服务可能无法对 Kerberos 密钥分发中心 (KDC) 进行身份验证。

如果您的其中一个 IdM 服务无法与另一个服务通信，请使用以下步骤验证您的 Kerberos keytab 文件是否与 IdM 数据库中存储的密钥同步。如果它们没有同步，请使用更新的密钥和 KVNO 检索 Kerberos keytab。这个示例比较并检索 IdM 服务器的更新 **DNS** 主体。

**先决条件**

- 您必须作为 IdM admin 帐户进行身份验证来检索 keytab 文件
- 您必须以 **root** 帐户身份验证来修改其他用户拥有的 keytab 文件

**流程**

1. 显示您要验证的 keytab 中的主体的 KVNO。在以下示例中，**/etc/named.keytab** 文件具有作为 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 IdM 数据库中存储的主体的 KVNO。在本例中，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，它的所有者为 **named** 用户。

```
[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
```

**其他资源**

- Identity Management 如何使用 Kerberos keytab 文件
- IdM Kerberos keytab 文件内容列表

## 16.3. IDM KERBEROS KEYTAB 文件内容列表

下表显示了 IdM Kerberos keytab 文件的位置、内容和目的。

**表 16.1. 表**

| keytab 位置 | 内容 | 用途 |
|---|---|---|
| **/etc/krb5.keytab** | **host** 主体 | 在登录时验证用户凭证，供 NFS 使用（如果没有 **nfs** 主体） |
| **/etc/dirsrv/ds.keytab** | **ldap** 主体 | 向 IdM 数据库验证用户，在 IdM 副本之间安全地复制数据库内容 |
| **/var/lib/ipa/gssproxy/http.keytab** | **HTTP** 主体 | 对 Apache 服务器进行身份验证 |
| **/etc/named.keytab** | **DNS** 主体 | 安全地更新 DNS 记录 |
| **/etc/ipa/dnssec/ipa-dnskeysyncd.keytab** | **ipa-dnskeysyncd** 主体 | 使 OpenDNSSEC 与 LDAP 同步 |
| **/etc/pki/pki-tomcat/dogtag.keytab** | **dogtag** 主体 | 与证书颁发机构 (CA) 通信。 |

| keytab 位置 | 内容 | 用途 |
| --- | --- | --- |
| **/etc/samba/samba.keytab** | **cifs** 和 **host** 主体 | 与 Samba 服务通信 |
| **/var/lib/sss/keytabs/*ad-domain.com*.keytab** | Active Directory (AD) 域控制器 (DC) 主体，格式为 ***HOSTNAME$@AD-DOMAIN.COM*** | 通过 IdM-AD Trust 与 AD DC 通信 |

**其他资源**

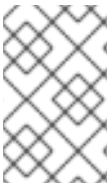
- Identity Management 如何使用 Kerberos keytab 文件
- 验证 Kerberos keytab 文件是否与 IdM 数据库同步

## 16.4. 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时用于加密它们的密钥。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 版本上使用的加密类型与遵循 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 部署系统兼容，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，因此即使 IdM 主密钥的加密类型是 **aes256-cts-hmac-sha384-192**，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 部署已安装在运行 RHEL 8.7 或更高版本的服务器上。

# 第 17 章 在 IDM 环境中启用 PASSKEY 身份验证

Fast IDentity Online 2 (FIDO2)标准基于公钥加密，并添加带有 PIN 或 biometrics 的免密码流的选项。IdM 环境中的 passkey 身份验证使用 **libfido2** 库支持的 FIDO2 兼容设备。

passkey 验证方法提供额外的安全层，通过包括需要 PIN 或指纹的免密码和多因素身份验证(MFA)来遵守法规标准。它使用特殊的硬件和软件（如 passkey 设备和 passkey 启用）在 Identity Management (IdM) 环境中增强安全性，以便在数据保护扮演关键角色的环境中增强安全性。

如果您的系统连接到 IdM 环境的网络，则 passkey 验证方法会自动发出 Kerberos 票据(SSO)，为 IdM 用户启用单点登录(SSO)。

您可以使用 passkey 通过图形界面向操作系统进行身份验证。如果您的系统允许您使用 passkey 和密码进行身份验证，则可以通过按键盘上的空格后跟 Enter 键来跳过 passkey 身份验证并 **使用密码进行身份验证。如果使用 GNOME 桌面管理器(GDM)，您可以按 Enter 来绕过 passkey 身份验证。**

**请注意，当前 IdM 环境中的 passkey 身份验证不支持 FIDO2 attestation 机制，它允许识别特定的 passkey 设备。**

**以下流程提供了在 IdM 环境中管理和配置 passkey 身份验证的说明。**

## 17.1. 先决条件

- **您有一个 passkey 设备。**
- **安装 fido2-tools 软件包：**

```
# dnf install fido2-tools
```

- 为 passkey 设备设置 PIN：
  1. **将 passkey 设备连接到 USB 端口。**
  2. **列出连接的 passkey 设备：**

```
# fido2-token -L
```

  3. **按照命令提示，为您的 passkey 设备设置 PIN。**

```
# fido2-token -C passkey_device
```

## 17.2. 注册 PASSKEY 设备

**作为用户，您可以使用 passkey 设备配置身份验证。passkey 设备与任何 FIDO2 规格设备兼容，如 YubiKey 5 NFC。要配置此验证方法，请按照以下说明操作。**

**先决条件**

- **设置 passkey 设备的 PIN。**
- 为 IdM 用户启用 Passkey 身份验证：

```
# ipa user-add user01 --first=user --last=01 --user-auth-type=passkey
```

将现有 IdM 用户的 ipa user-mod 与相同的 --user-auth-type=passkey 参数一起使用。

- 访问用户要进行身份验证的物理计算机。

流程

1. 在 USB 端口中插入 passkey 设备。
2. 为 IdM 用户注册 passkey：

```
# ipa user-add-passkey user01 --register
```

按照应用程序提示：

a. 输入 passkey 设备的 PIN。

b. 接触设备以验证您的身份。如果您使用 biometric 设备，请确保使用与注册该设备相同的 finger。

用户最好将多个 passkey 设备配置为允许从多个位置或设备进行身份验证的备份。要确保在身份验证过程中发布 Kerberos 票据，请不要为用户配置超过 12 个 passkey 设备。

验证

1. 使用您配置为使用 passkey 身份验证的用户名登录到系统。系统会提示您插入 passkey 设备：

```
Insert your passkey device, then press ENTER.
```

2. 将 passkey 设备插入到 USB 端口中，并在提示时输入您的 PIN：

```
Enter PIN:
Creating home directory for user01@example.com.
```

3. 确认已发布 Kerberos 票据：

```
$ klist
Default principal: user01@IPA.EXAMPLE.COM
```

请注意，要跳过 passkey 身份验证，请在提示符中输入任意字符，或者在启用了用户身份验证时输入空 PIN。系统会将您重定向到基于密码的身份验证。

## 17.3. 身份验证策略

使用身份验证策略配置可用的在线和本地身份验证方法。

使用在线连接进行身份验证

使用服务在服务器端提供的所有在线身份验证方法。对于 IdM、AD 或 Kerberos 服务，默认的验证方法是 Kerberos。

在没有在线连接的情况下进行身份验证

使用可供用户使用的身份验证方法。您可以使用 local_auth_policy 选项调整身份验证方法。

使用 /etc/sssd/sssd.conf 文件中的 local_auth_policy 选项配置可用的在线和离线身份验证方法。默认情况下，只有使用服务服务器端支持的方法来执行身份验证。您可以使用以下值调整策略：

- match 值启用与离线和在线状态匹配的。例如，IdM 服务器支持在线 passkey 身份验证，并匹配为 passkey 方法启用离线和在线身份验证。
- 唯一值仅提供 离线方法并忽略在线方法。
- 启用和禁用 值明确定义了用于离线身份验证的方法。例如，enable:passkey 只启用 passkey 进行离线身份验证。

以下配置示例允许本地用户使用智能卡验证在本地进行身份验证：

```
[domain/shadowutils]
id_provider = proxy
proxy_lib_name = files
auth_provider = none
local_auth_policy = only
```

local_auth_policy 选项适用于 passkey 和智能卡身份验证方法。

## 17.4. 以 PASSKEY 用户身份检索 IDM TICKET-GRANTING TICKET

要以 passkey 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 9.1 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。
- 您已注册了 passkey 设备并配置了身份验证策略。

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 PIN:
```

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) = 153
```

**pa_type = 153** 表示 passkey 身份验证。

# 第 18 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。

## 18.1. 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
```

## 18.2. 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。

## 18.3. 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
```

## 18.4. 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
```

## 18.5. 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
```

其他资源

- 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。

## 18.6. 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
```

**其他资源**

- 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 19 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。

## 19.1. 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。

## 19.2. 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。

**步骤**

- 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
```

## 19.3. 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)在 IdM 中编辑自助服务访问规则。

**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
```

## 19.4. 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)在 IdM 中删除自助服务访问规则。

**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。

步骤

- 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 20 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。

## 20.1. 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。

## 20.2. 使用 IDM WEB UI 创建自助服务规则

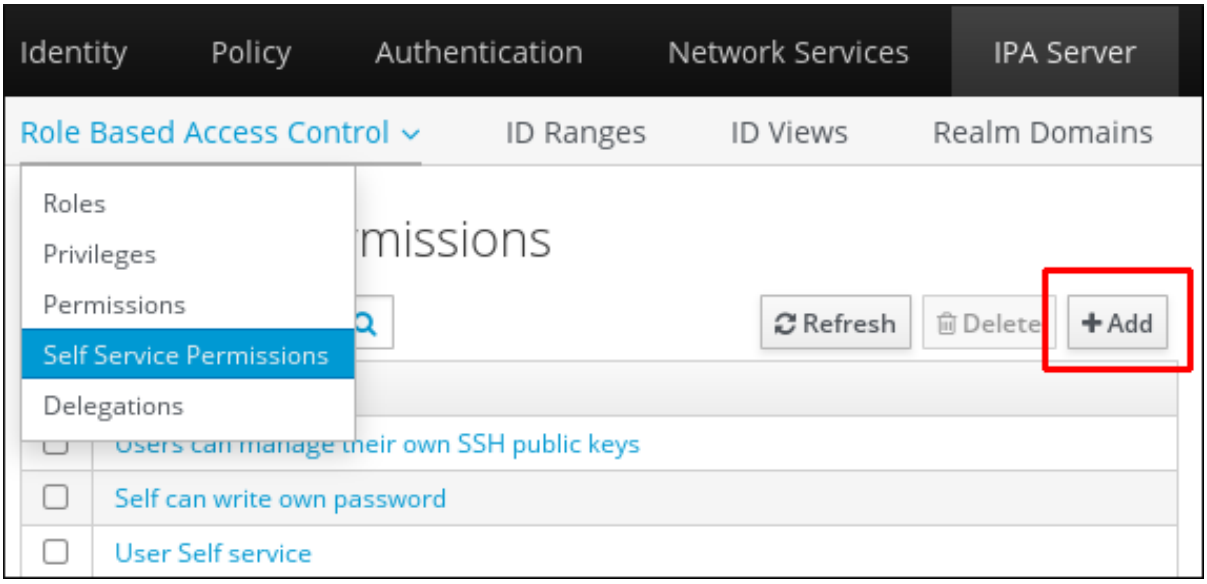
按照以下流程，使用 Web 界面(IdM Web UI)在 IdM 中创建自助服务访问规则。

**先决条件**

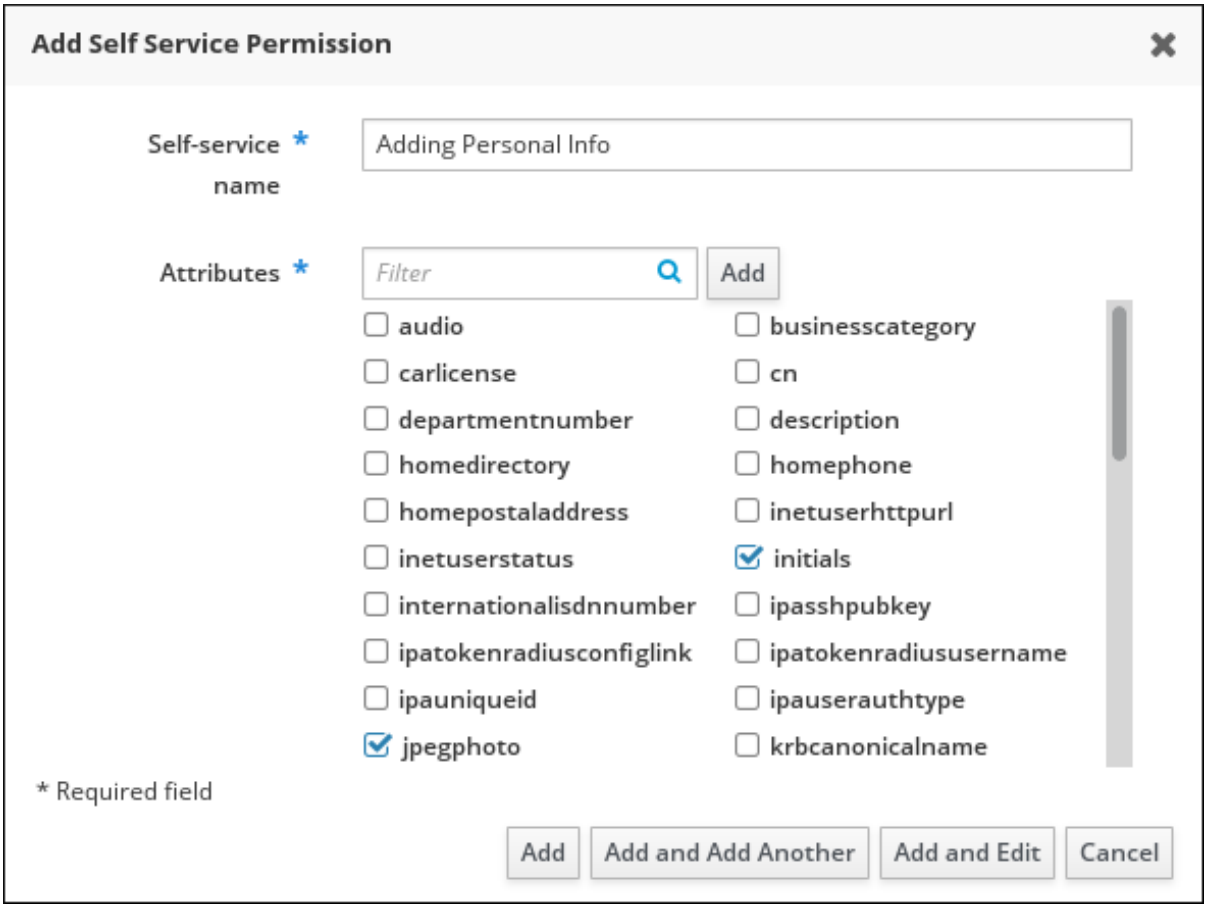
- 管理 IdM 或 用户管理员 角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

**步骤**

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。

## 20.3. 使用 IDM WEB UI 编辑自助服务规则

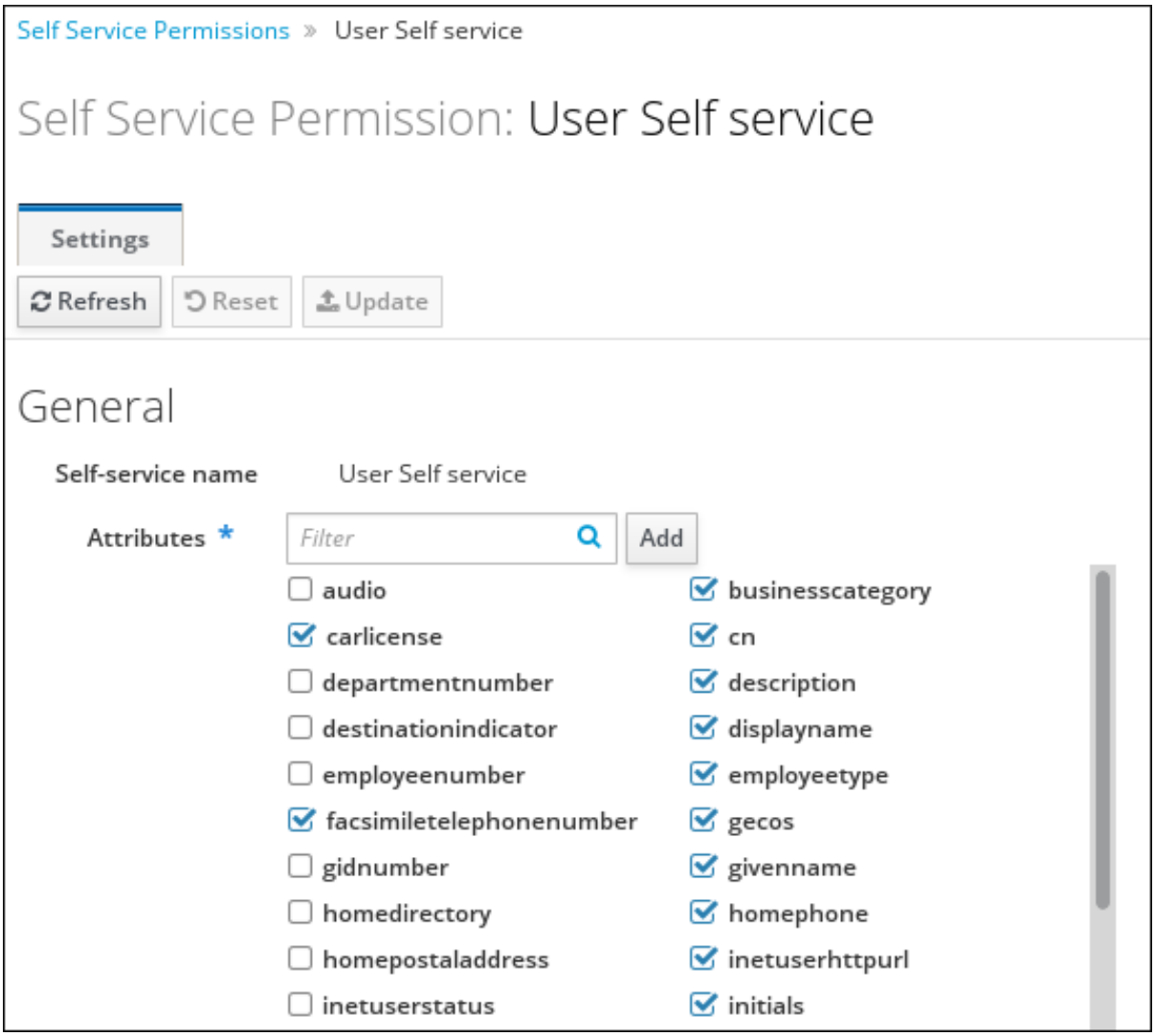
按照以下流程，使用 Web 界面(IdM Web UI)在 IdM 中编辑自助服务访问规则。

**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。

**步骤**

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。

## 20.4. 使用 IDM WEB UI 删除自助服务规则

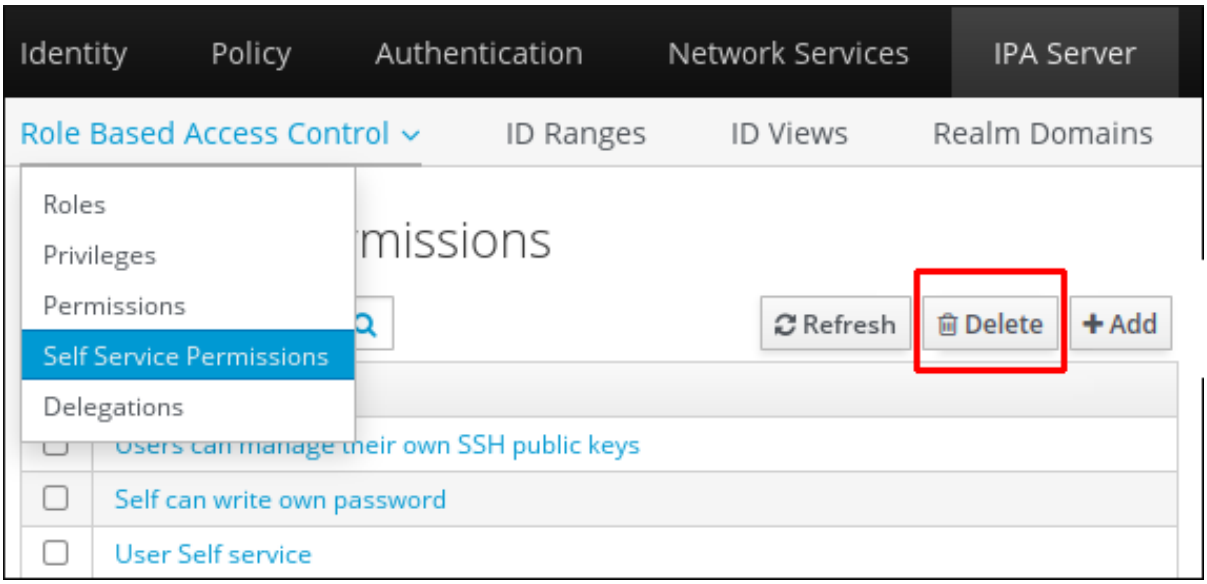
**按照以下流程，使用 Web 界面(IdM Web UI)删除 IdM 中的自助服务访问规则。**

**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。**

**步骤**

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 21 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

## 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 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

步骤

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 权限 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. 保存这个文件。

6. 运行 Ansible 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 inventory selfservice-present-copy.yml
```

**其他资源**

- 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。

## 21.3. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

**步骤**

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. 保存这个文件。
6. 运行 Ansible 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 inventory selfservice-
absent-copy.yml
```

**其他资源**

- 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。

## 21.4. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。
- Users can manage their own name details自助服务规则存在于 IdM 中。

步骤

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. 保存这个文件。
6. 运行 Ansible 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 inventory selfservice-member-present-copy.yml
```

**其他资源**

- 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。

## 21.5. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- Users can manage their own name details自助服务规则存在于 IdM 中。

**步骤**

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. 保存这个文件。
6. 运行 Ansible 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 inventory selfservice-
member-absent-copy.yml
```

**其他资源**

- 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 22 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

## 22.1. 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 22.1. 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。

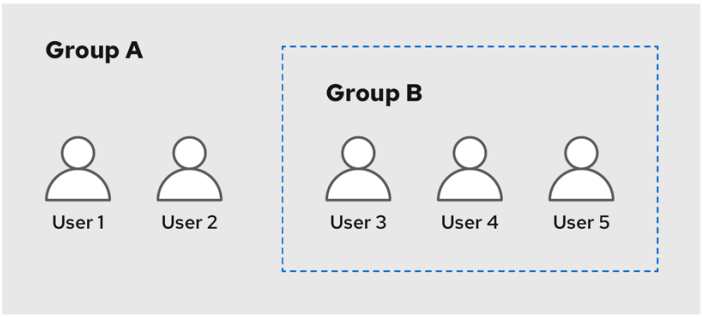
## 22.2. 直接和间接组成员

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 22.1. 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。

## 22.3. 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。

**步骤**

- 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。

## 22.4. 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。

步骤

- 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。

## 22.5. 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。

步骤

- 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"
-------------------------
```

## 22.6. 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 和 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。

步骤

- 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。

## 22.7. 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 。

### 22.7.1. 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。

### 22.7.2. 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。

步骤

- 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
```

### 22.7.3. 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。

步骤

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
```

### 22.7.4. 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**

**步骤**

- **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。**

## 22.8. 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。**

**步骤**

- **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：**

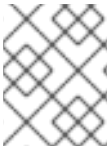
```
$ ipa group-add-member-manager group_a --users=test
Group name: group_a
GID: 1133400009
Membership managed by users: test
-------------------------
Number of members added 1
-------------------------
```

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
```

**其他资源**

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

## 22.9. 使用 IDM CLI 查看组成员

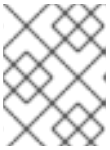
按照以下流程，使用 IdM CLI 查看组成员。您可以查看直接和间接组成员。如需更多信息，请参阅 直接和间接组成员。

**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。

## 22.10. 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。

**步骤**

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
```

## 22.11. 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 从 IdM 用户组中删除作为成员管理者的用户或组。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。

**步骤**

- 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
```

其他资源

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

## 22.12. 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 活动目录(AD)）提供，或者它们本地系统上的 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。

如果要从域控制器和本地 etc/group 文件管理组，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同时出现，则成员用户列表被合并，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。

步骤

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：

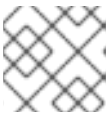
```
# ipa user-add idmuser
First name: idm
Last name: user
---------------------
Added user "idmuser"
---------------------
User login: idmuser
First name: idm
Last name: user
Full name: idm user
Display name: idm user
Initials: tu
Home directory: /home/idmuser
GECOS: idm user
Login shell: /bin/sh
Principal 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 中时定义的 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)
```

## 22.13. 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 9.4 或更高版本。**
  - **示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个带有 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 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,也就是在其上执行 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。

步骤

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、存储保护 secret.yml 文件的密码，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)
```

**其他资源**

- 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 23 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

## 23.1. 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 23.1. 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。

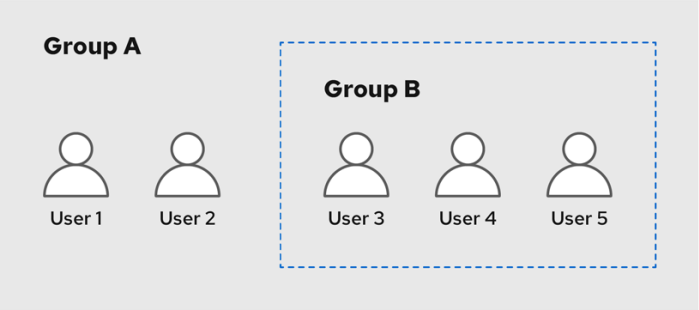
## 23.2. 直接和间接组成员

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 23.1. 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。

## 23.3. 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。

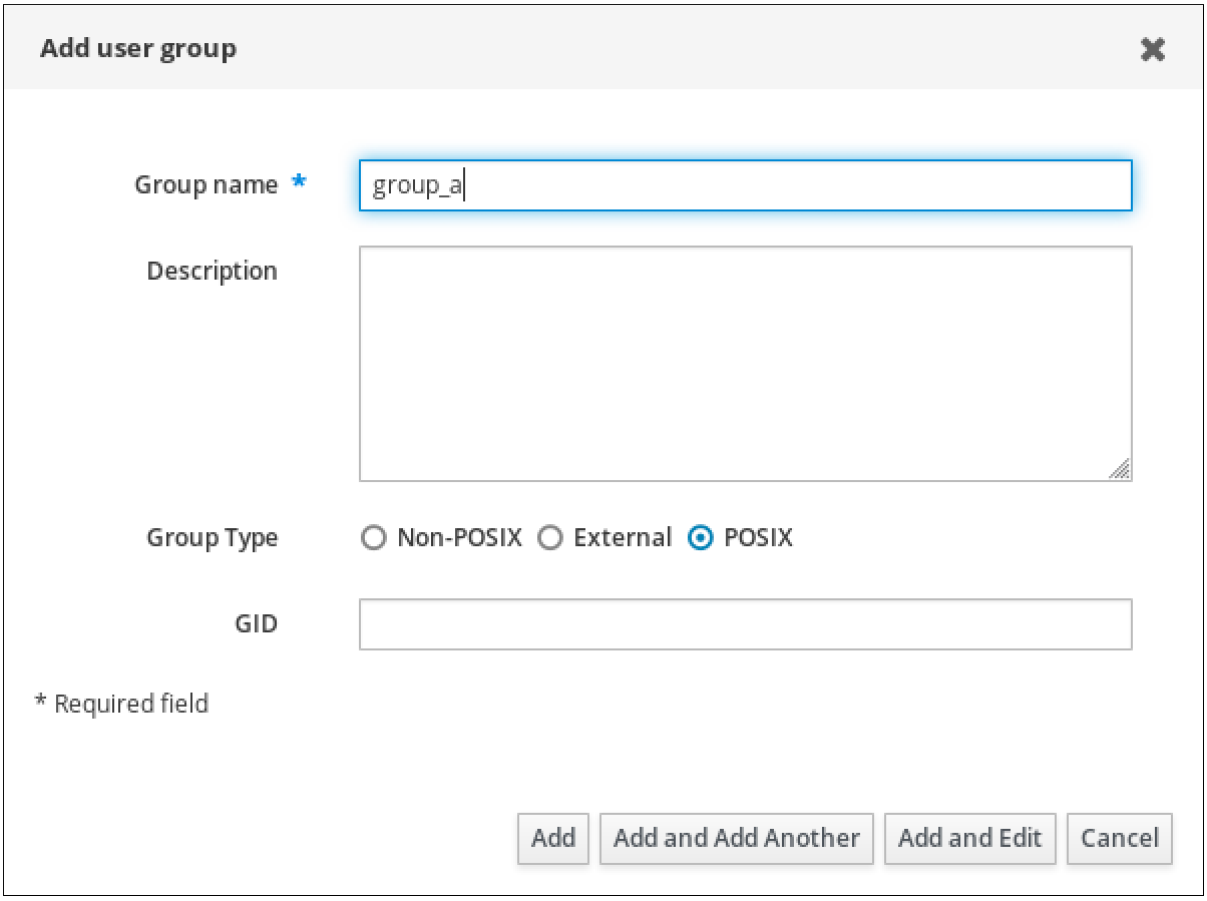
**步骤**

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。

### 23.4. 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。

**步骤**

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。

### 23.5. 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**

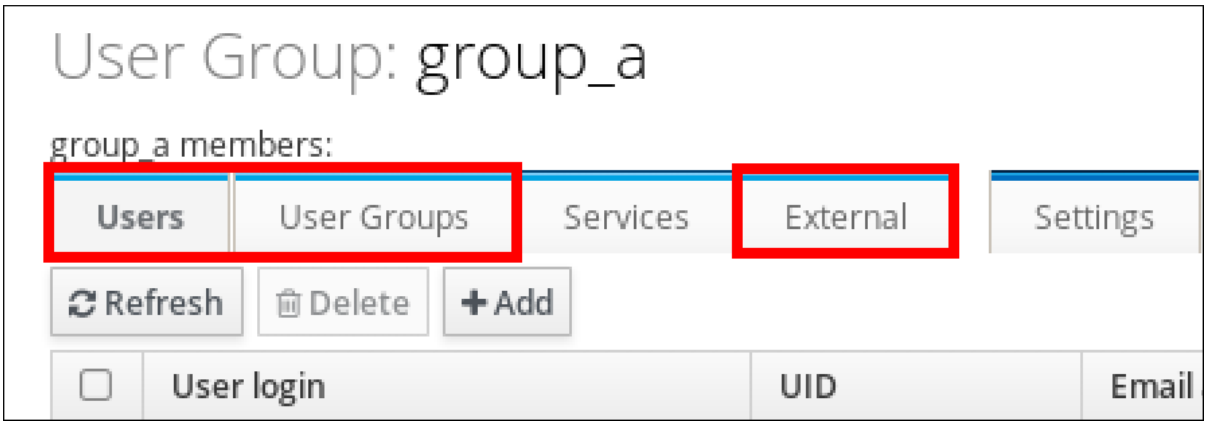
- 已登陆到 IdM Web UI。

**步骤**

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

2. 单击组的名称。

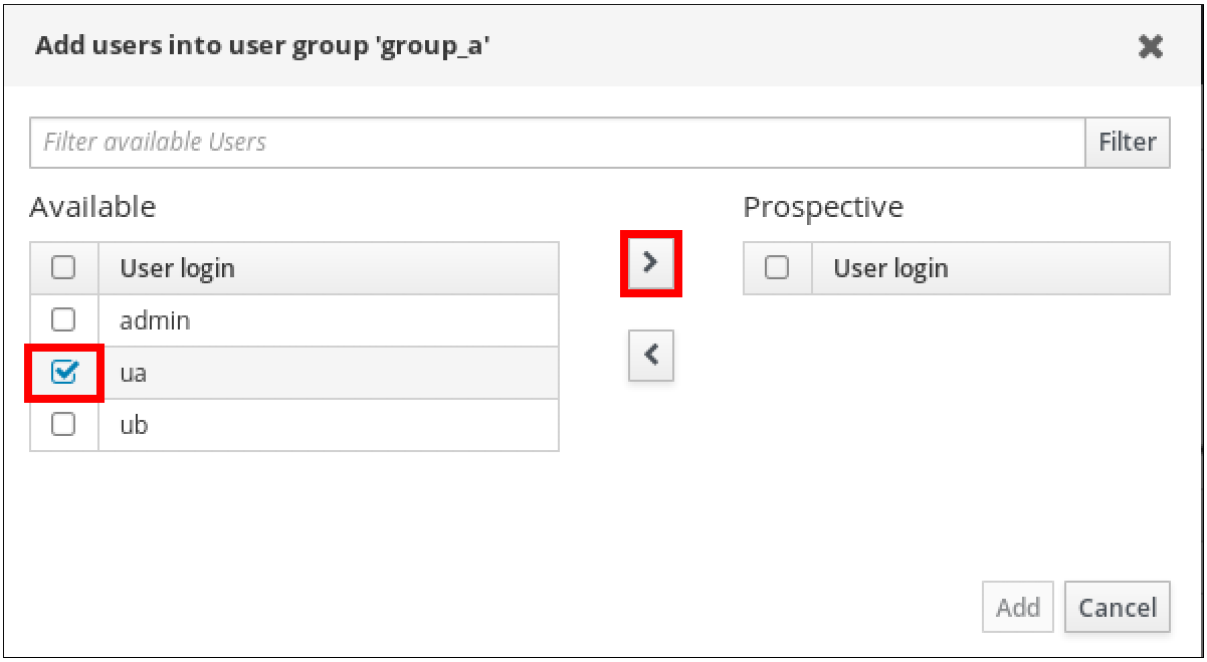
3. 选择您要添加的组成员的类型： **User**、**User Groups** 或 **External**。



4. 点 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。

## 23.6. 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 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**

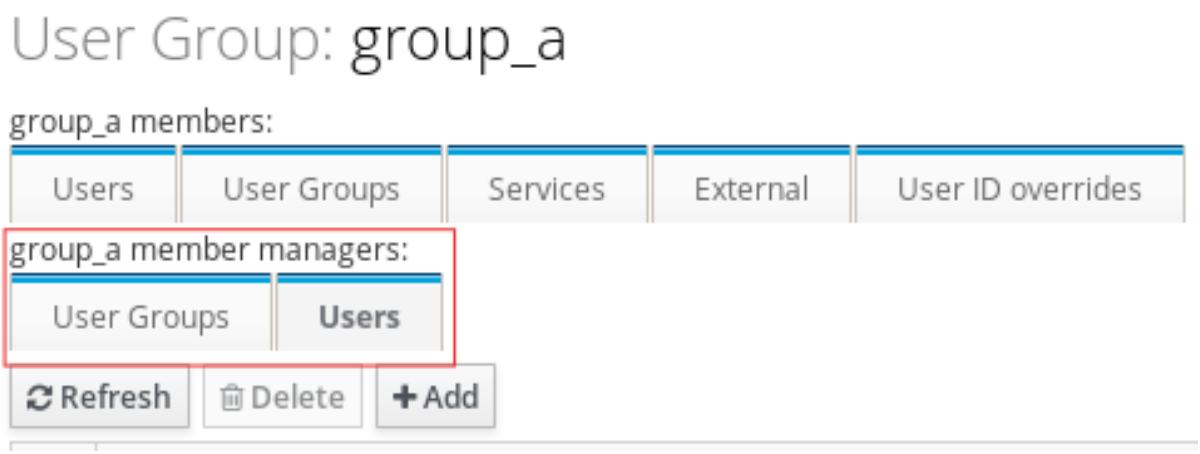
- 已登陆到 IdM Web UI。
- 您必须具有要添加为成员管理器的用户名以及您要管理的组的名称。

**步骤**

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

2. 单击组的名称。

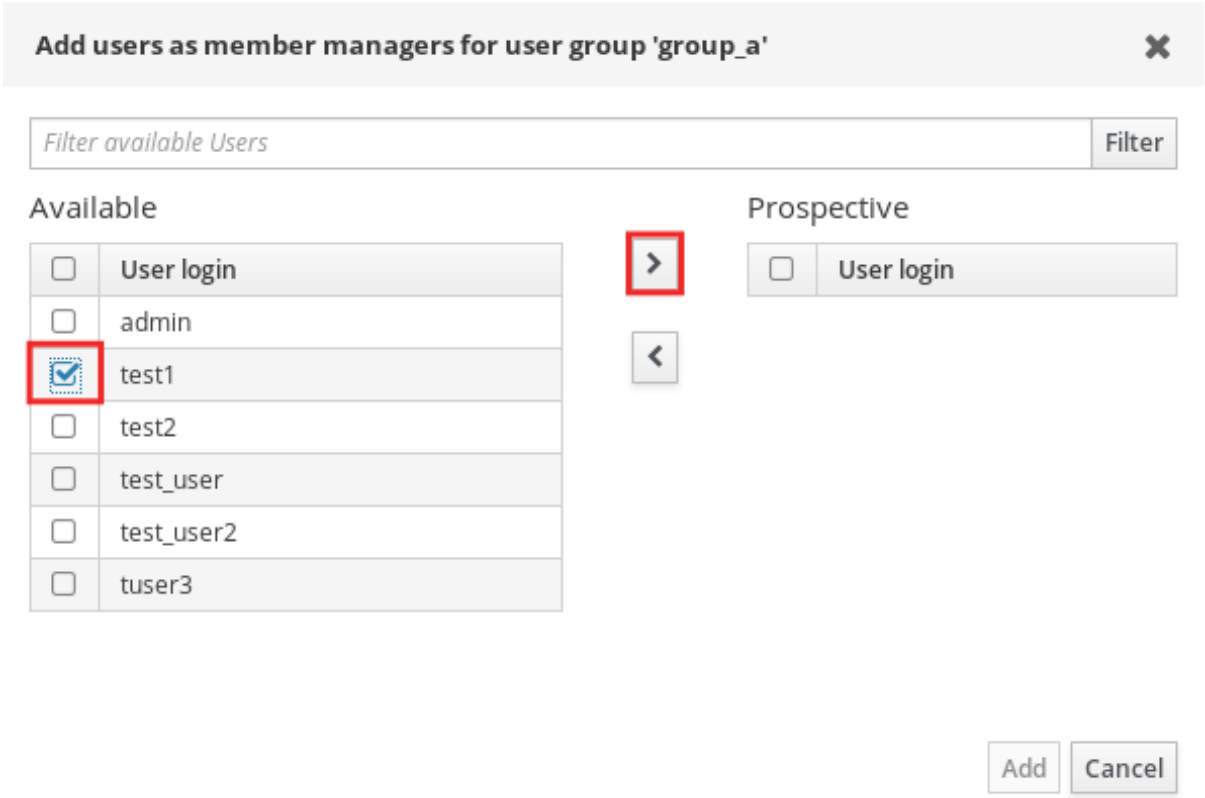
3. 选择您要添加的组成员管理者的类型：**Users** 或 **User Groups**。




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。



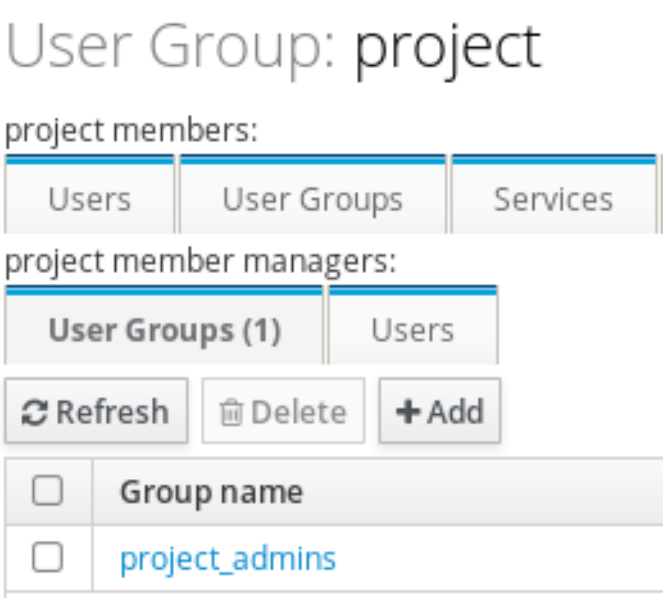

7. 单击 **Add** 确认。

**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：




**其他资源**

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

### 23.7. 使用 IDM WEB UI 查看组成员

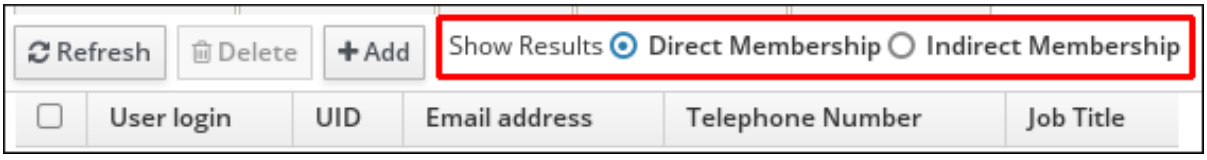
按照以下流程，使用 IdM Web UI 查看组成员。您可以查看直接和间接组成员。如需更多信息，请参阅直接和间接组成员。

**先决条件**

- 已登陆到 IdM Web UI。

**步骤**

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。




### 23.8. 使用 IDM WEB UI 从用户组中删除成员

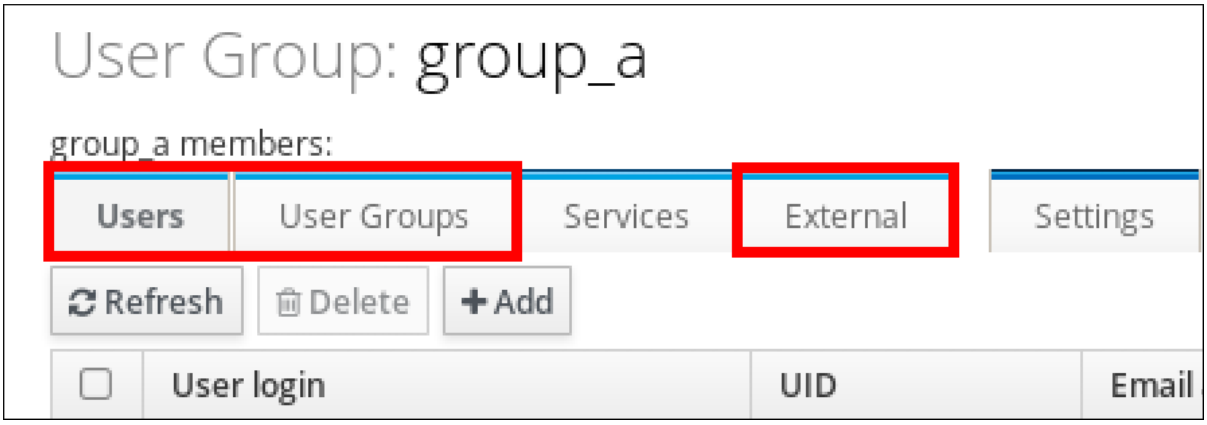
**按照以下流程，使用 IdM Web UI 从用户组中删除成员。**

**先决条件**

- **已登陆到 IdM Web UI。**

**步骤**

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。**

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

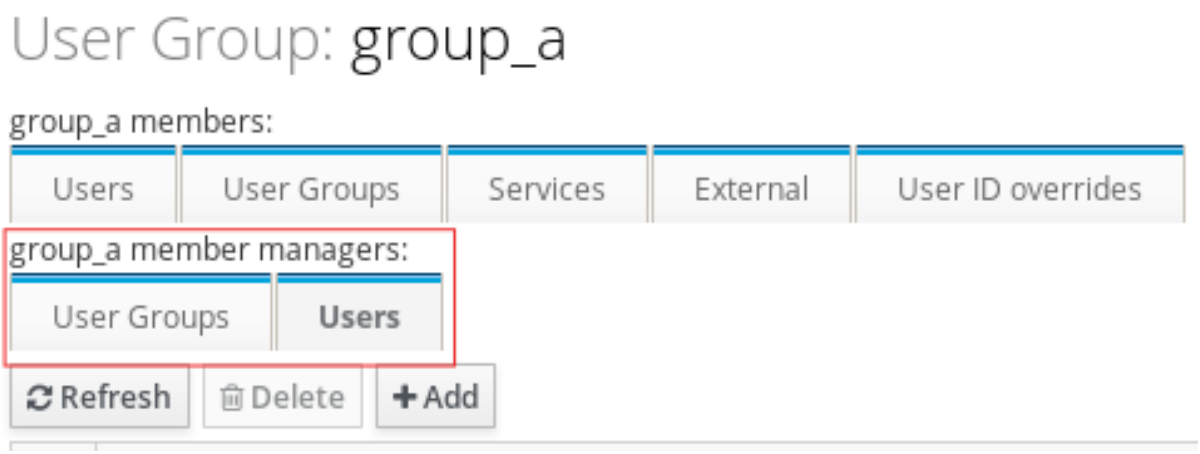
按照以下流程，使用 Web UI 从 IdM 用户组中删除作为成员管理者的用户或组。成员管理者可以从 IdM 用户组中删除用户或组，但不能更改组的属性。

**先决条件**

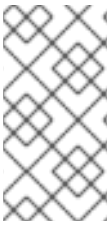
- 已登陆到 IdM Web UI。
- 您必须具有要删除的现有成员管理者用户或组的名称，以及它们要管理的组的名称。

**步骤**

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。

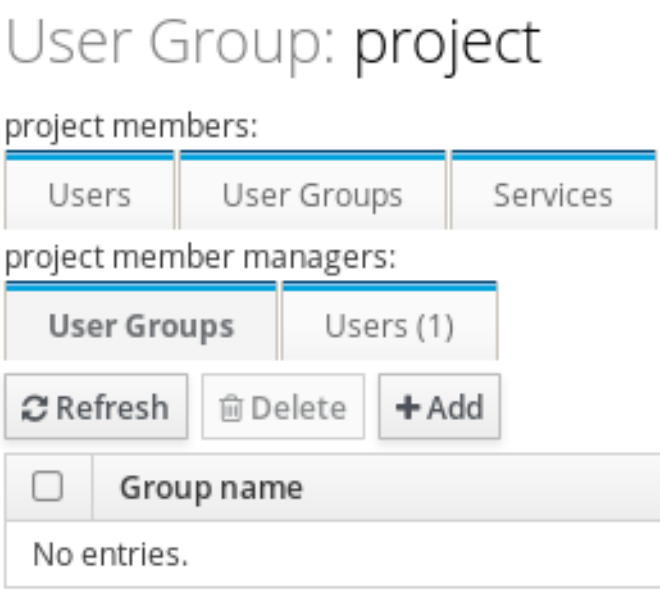


**注意**

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：




**其他资源**

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

# 第 24 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

## 24.1. 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 24.1. 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。

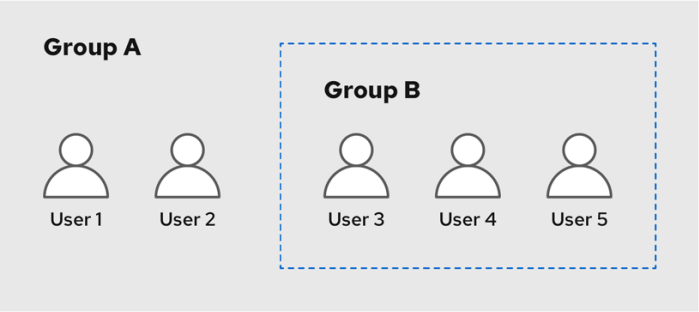
## 24.2. 直接和间接组成员

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 24.1. 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。

## 24.3. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
-

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。

步骤

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。

其他资源

- 请参阅 /usr/share/doc/ansible-freeipa/README-group.md Markdown 文件。

### 24.4. 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，使用单个 Ansible 任务添加、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，并将 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*/ 目录中创建了一个带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您在使用 RHEL 9.3 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
```

其他资源

- ansible-freeipa 上游 docs 中的组模块

## 24.5. 使用 ANSIBLE 启用 AD 用户管理 IDM

按照以下流程，使用 Ansible playbook 确保用户 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。
- 您要添加用户 ID 覆盖的组已在 IdM 中存在。
- 您可以使用 IdM 或更高版本的 4.8.7 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

**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 admin 密码。

- admins 是您要添加 ad_user@ad.example.com ID 覆盖的 IdM POSIX 组的名称。此组中的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、存储保护 secret.yml 文件的密码，以及清单文件：

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

其他资源

- 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

## 24.6. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

- 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。

**步骤**

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
```

其他资源

- 请参阅 ipa host-add-member-manager --help。
- 请参阅 ipa man page。

### 24.7. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- 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。

步骤

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
```

其他资源

- 请参阅 ipa host-remove-member-manager --help。
- 请参阅 ipa man page。

# 第 25 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：


- 自动化组成员资格的好处
- 自动成员规则
- 使用 IdM CLI 添加自动成员规则
- 使用 IdM CLI 将条件添加到自动成员规则中
- 使用 IdM CLI 查看现有的自动成员规则
- 使用 IdM CLI 删除自动成员规则
- 使用 IdM CLI 从自动成员规则中删除条件
- 使用 IdM CLI 将自动成员规则应用到现有条目

- 使用 IdM CLI 配置默认的自动成员组

## 25.1. 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。

## 25.2. 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：

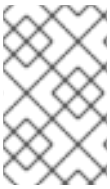
- 包含的条件

  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。

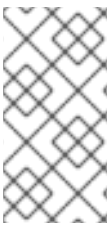
自动成员规则适用于将来创建的每个条目。这些条目将自动添加到指定的目标组中。如果一个条目满足多个自动成员规则中指定的条件，它将被添加到所有对应的组中。

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。

### 25.3. 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。

**步骤**

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。

### 25.4. 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。

步骤

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。

### 25.5. 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。

步骤

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
----------------------------
```

## 25.6. 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。

### 步骤

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 。

## 25.7. 使用 IDM CLI 从自动成员规则中删除条件

按照以下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。

步骤

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。

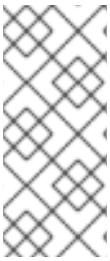
例如：

```
$ 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
-----------------------------
```

## 25.8. 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。详情请查看链接：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。

## 25.9. 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**

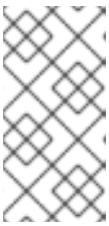
- 您必须以管理员身份登录。详情请参阅 使用 kinit 手动登录到 IdM。
- 您要设置为默认的目标组在 IdM 中已存在。

**步骤**

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 26 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：

- 自动化组成员资格的好处
- 自动成员规则
- 使用 IdM Web UI 添加自动成员规则
- 使用 IdM Web UI 向自动成员规则中添加条件
- 使用 IdM Web UI 查看现有的自动成员规则和条件
- 使用 IdM Web UI 删除自动成员规则
- 使用 IdM Web UI 从自动成员规则中删除条件
- 使用 IdM Web UI 将自动成员规则应用到现有条目

- 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

## 26.1. 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。

## 26.2. 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

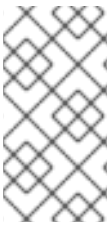
IdM 在包含条件之前评估排他条件。在发生冲突时，排他条件优先于包含条件。

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。

### 26.3. 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。

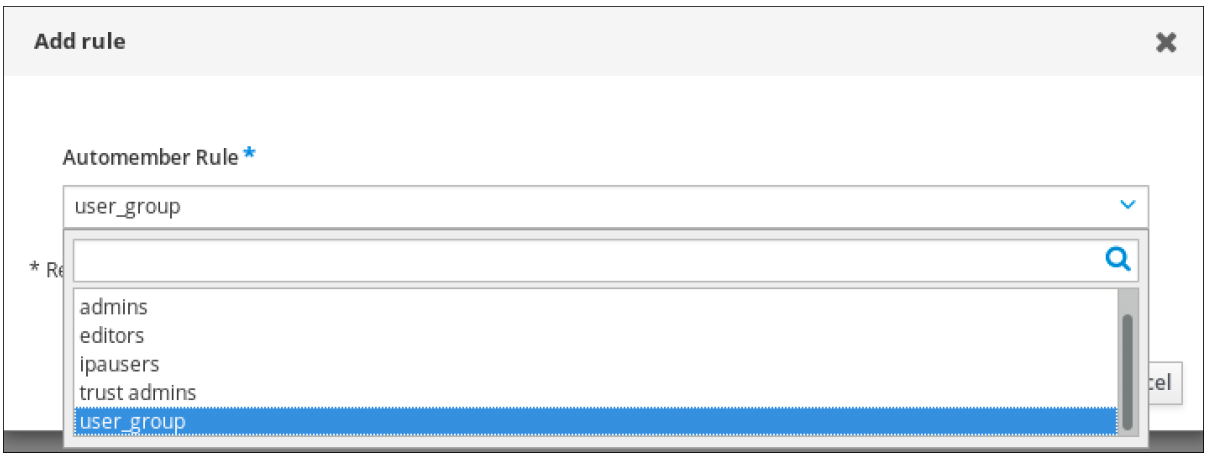
先决条件

- 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。

**步骤**

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。

## 26.4. 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。

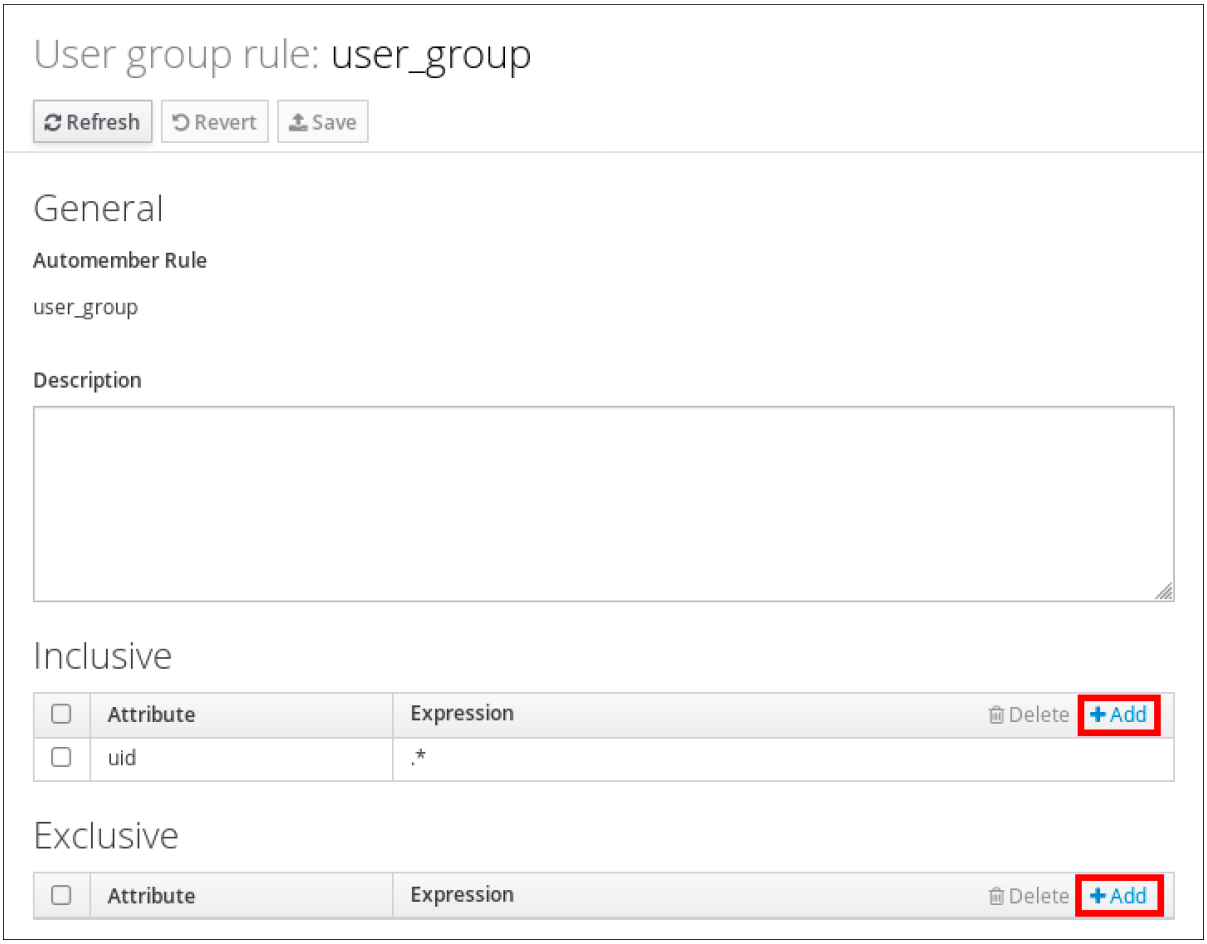
**先决条件**

- 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。

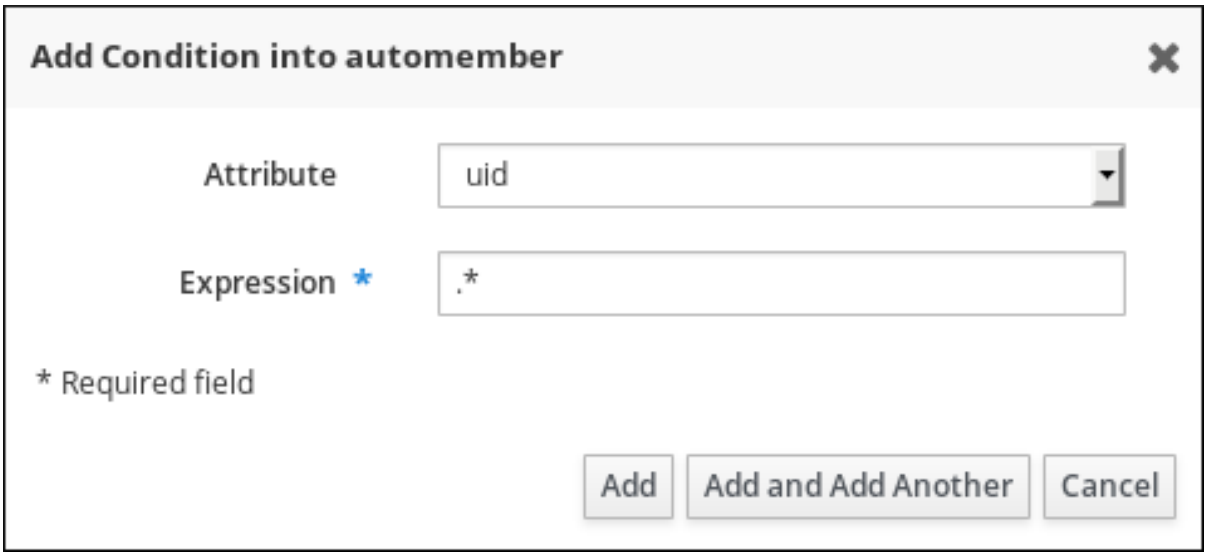
**步骤**

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。




## 26.5. 使用 IDM WEB UI 查看现有的自动成员规则和条件

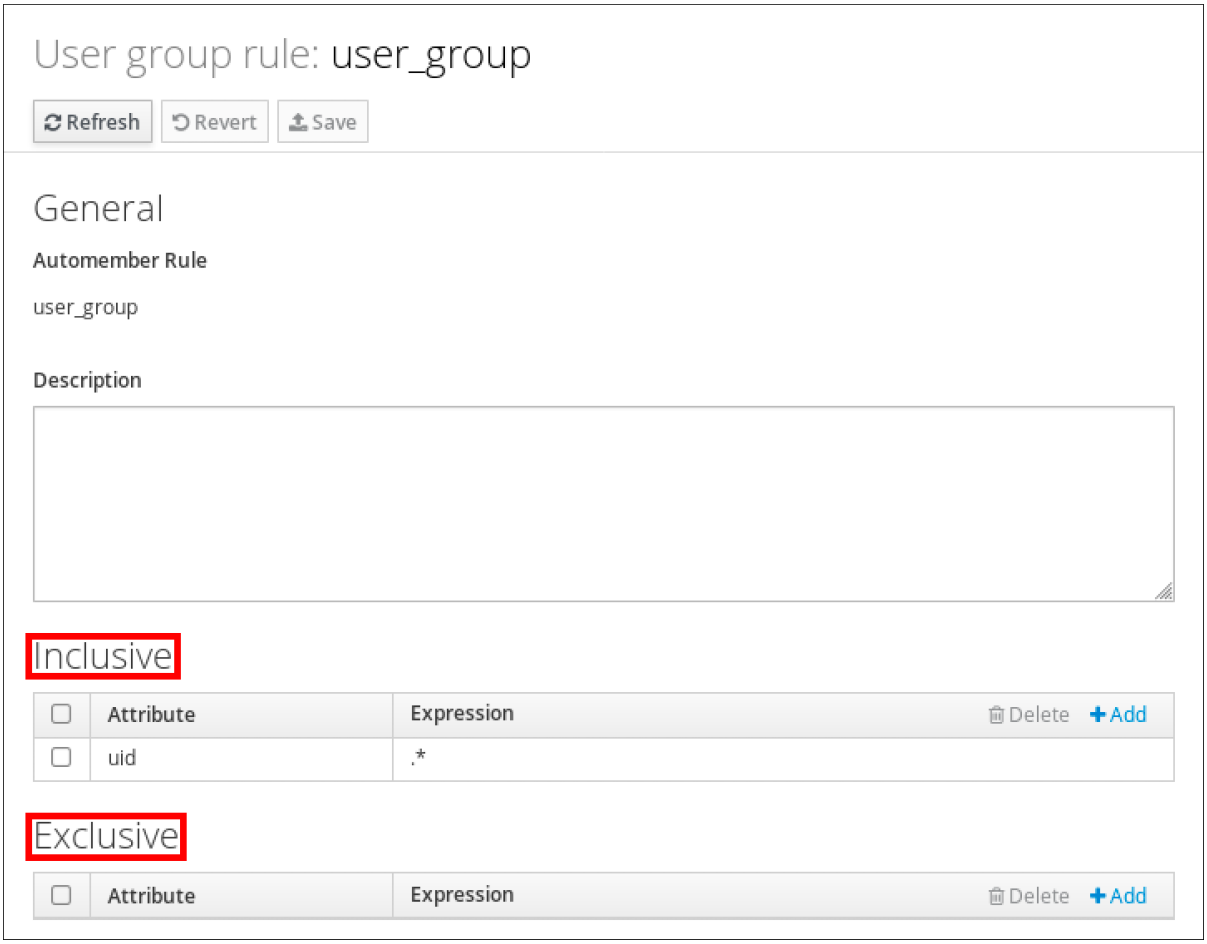
按照以下流程，使用 IdM Web UI 查看现有的自动成员规则和条件。

**先决条件**

- 已登陆到 IdM Web UI。
- 您必须是 admins 组的成员。

**步骤**

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。




## 26.6. 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

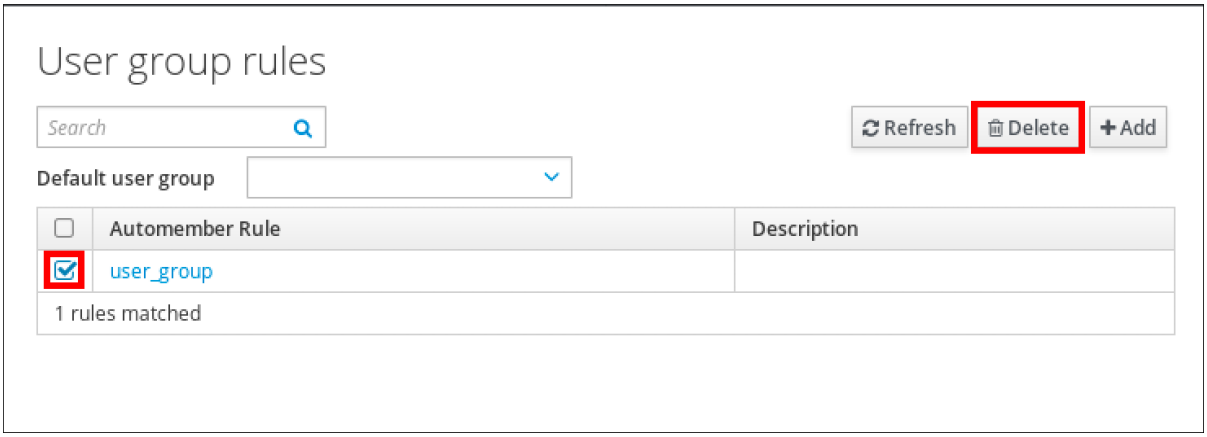
- 已登陆到 IdM Web UI。
- 您必须是 admins 组的成员。

### 步骤

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。

## 26.7. 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。

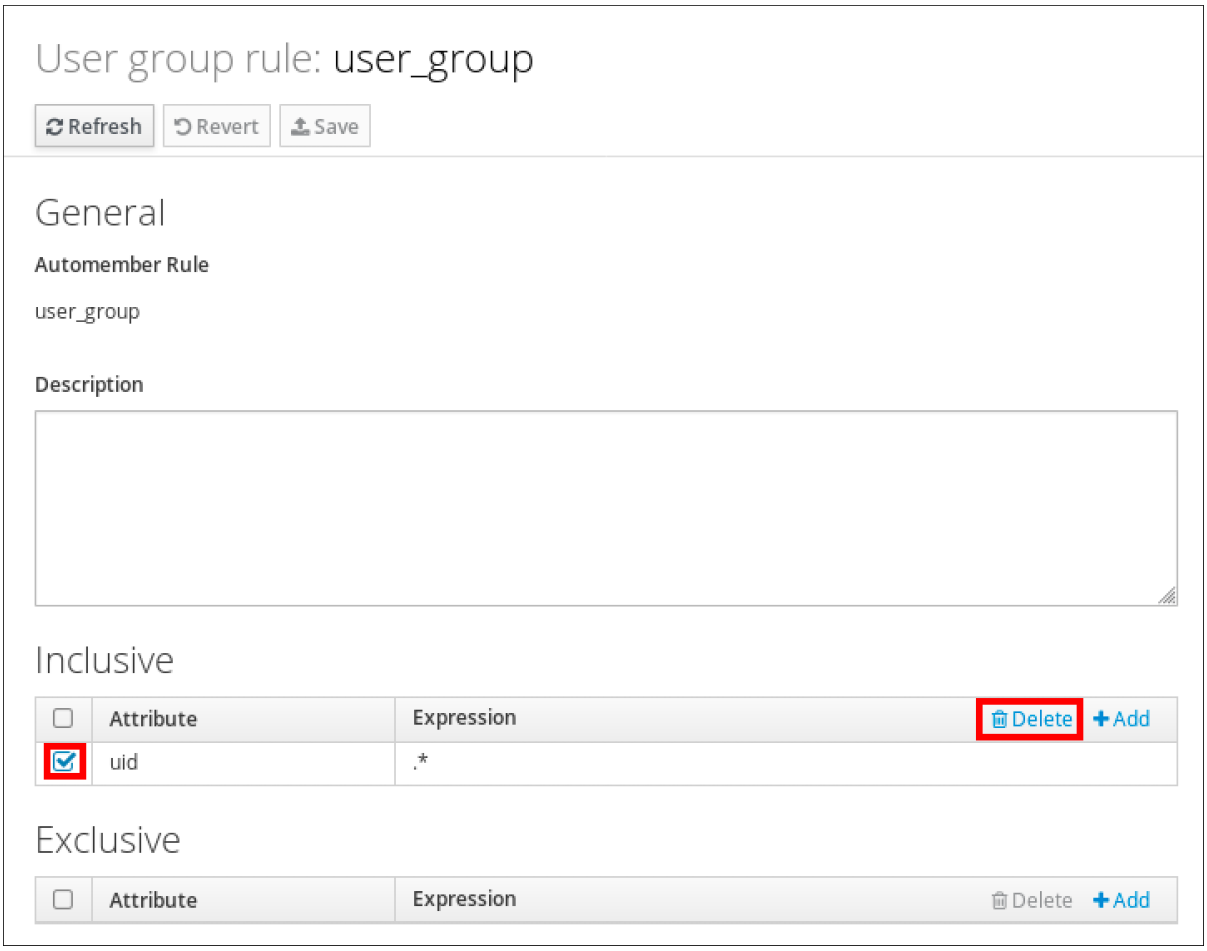
### 步骤

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**。

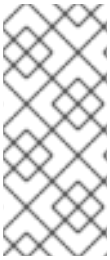



5. 单击 **Delete** 确认。

## 26.8. 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。

### 26.8.1. 为所有用户或主机重建自动成员资格

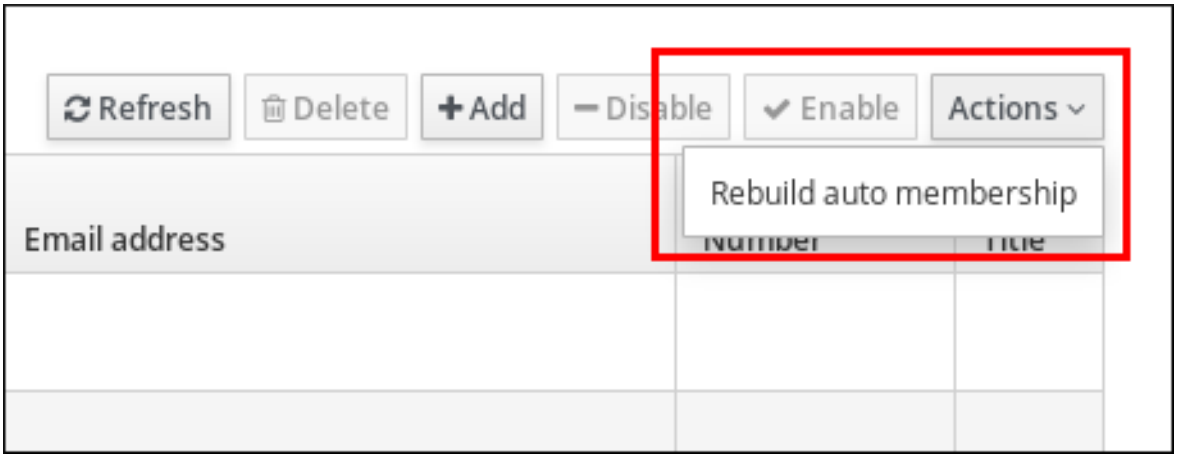
**按照以下流程，为所有用户或主机条目重建自动成员资格。**

**先决条件**

- **已登陆到 IdM Web UI。**
- **您必须是 admins 组的成员。**

**步骤**

1. 选择 **Identity → Users 或 Hosts。**
2. 单击 **Actions → Rebuild auto membership。**




### 26.8.2. 只为单个用户或主机重建自动成员资格

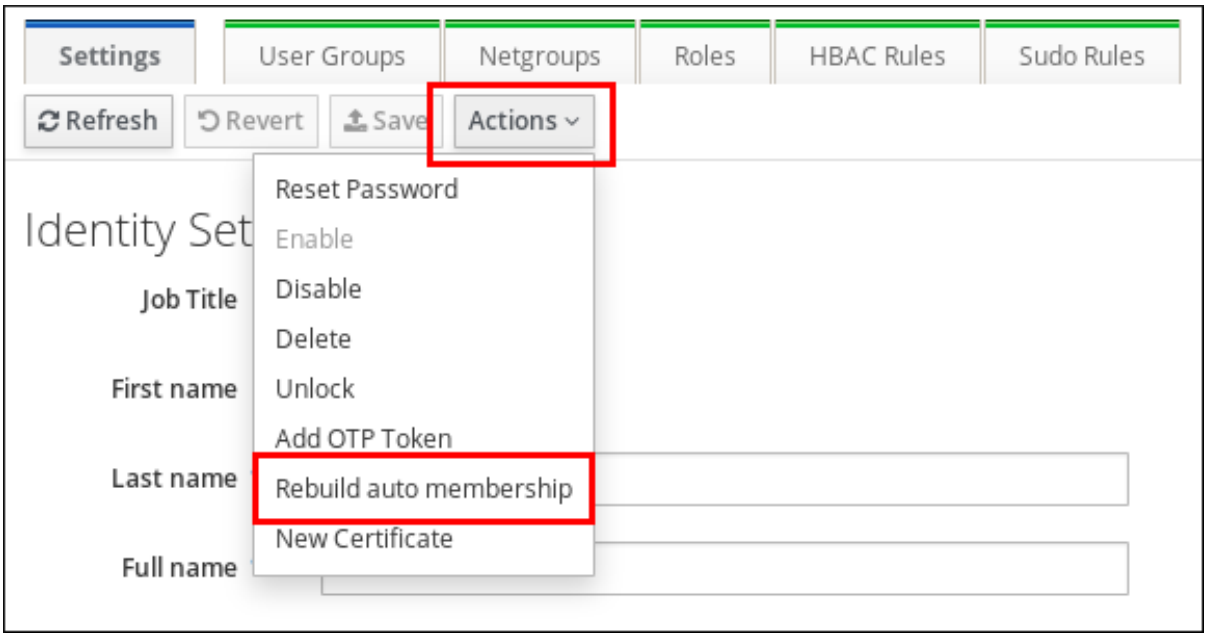
**按照以下流程，为特定用户或主机条目重建自动成员资格。**

**先决条件**

- **已登陆到 IdM Web UI。**
- **您必须是 admins 组的成员。**

**步骤**

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**。




## 26.9. 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。**

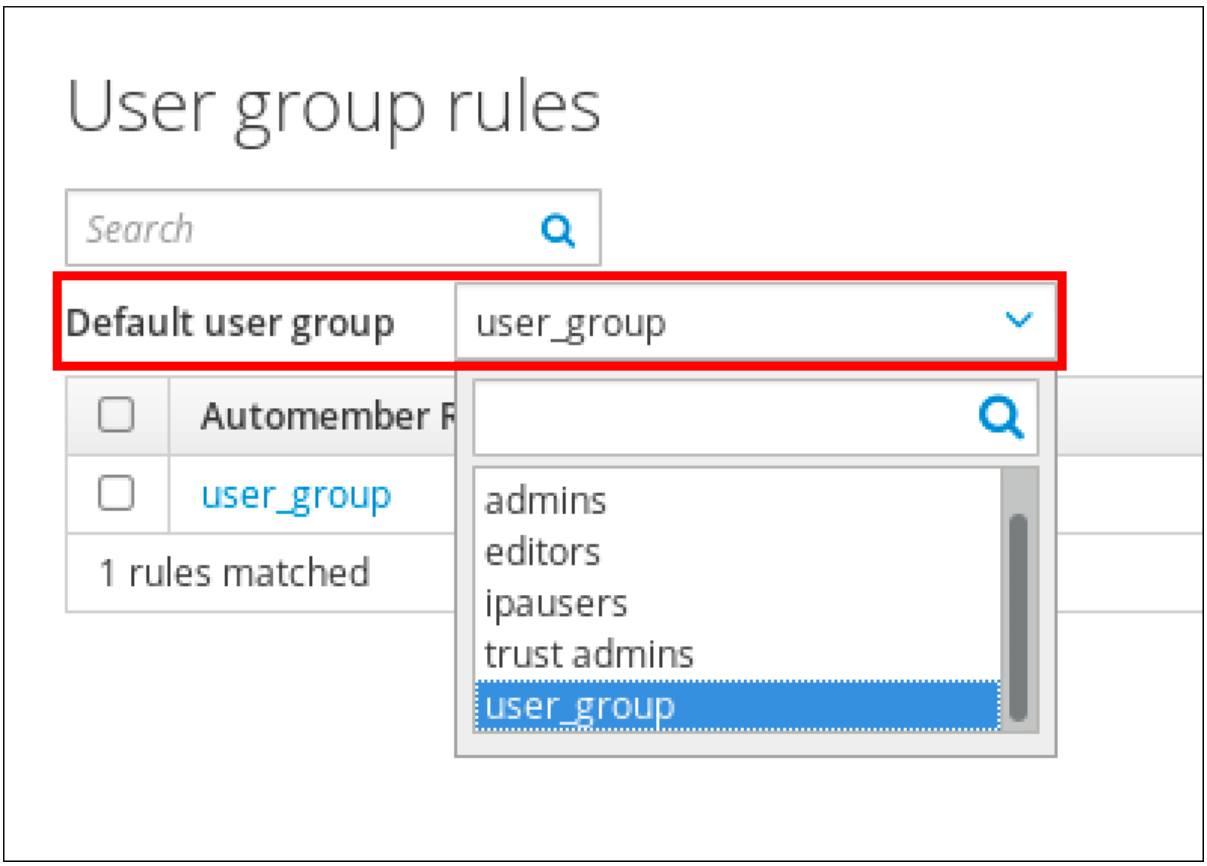
**先决条件**

- **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。**

**步骤**

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。




## 26.10. 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。

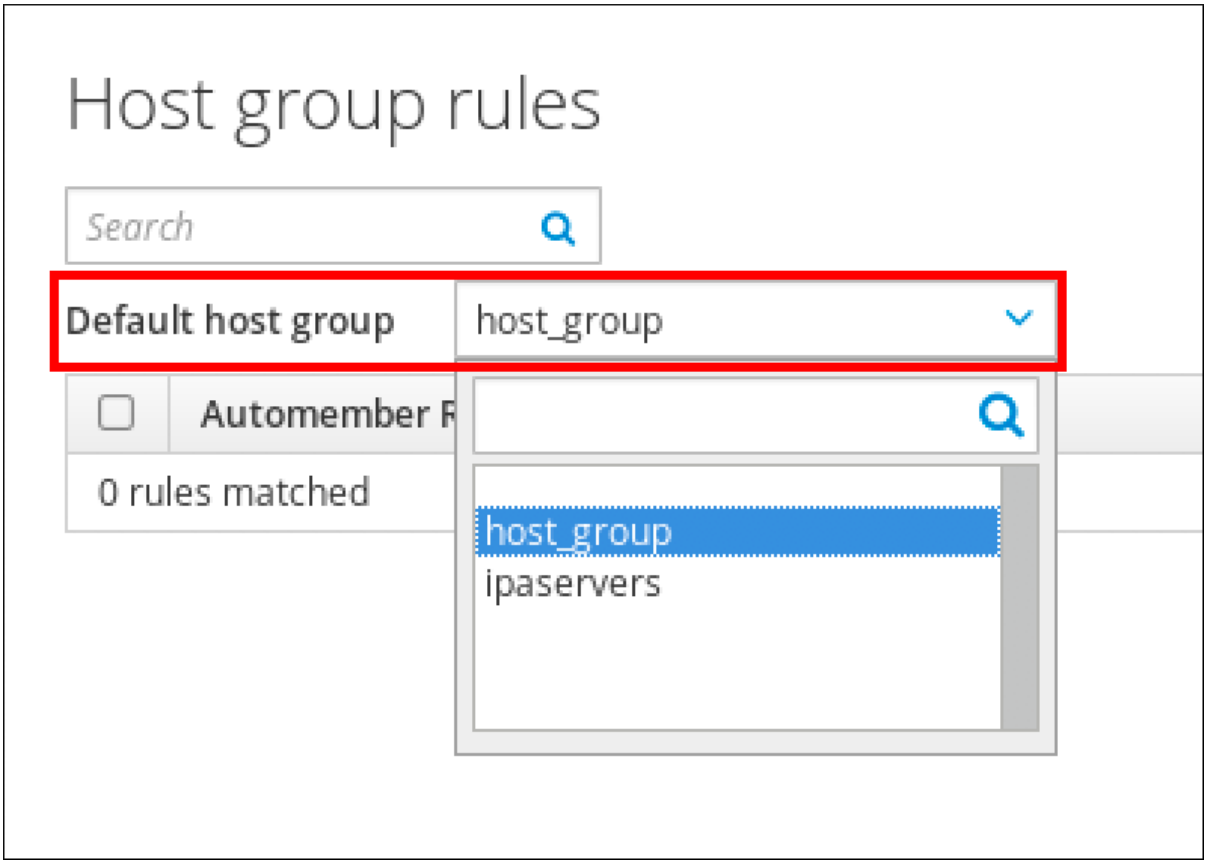
**先决条件**

- 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。

**步骤**

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 27 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：

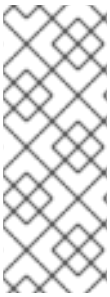
- 准备 Ansible 控制节点来管理 IdM
- 使用 Ansible 确保 IdM 用户组的自动成员规则存在
- 使用 Ansible 确保 IdM 用户组自动成员规则中存在条件
- 使用 Ansible 确保 IdM 用户组自动成员规则中的条件不存在
- 使用 Ansible 确保 IdM 组的自动成员规则不存在
- 使用 Ansible 确保 IdM 主机组自动成员规则中存在条件

## 27.1. 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 /usr/share/doc/ansible-freeipa/* 和 /usr/share/doc/rhel-system-roles/* 目录和子目录中的示例 Ansible playbook 复制到 ~/MyPlaybooks 目录中。
- 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。

步骤

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。

**其他资源**

- 使用 Ansible playbook 安装身份管理服务器。
- 如何构建清单。

## 27.2. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 inventory automember-group-present-copy.yml
```

**其他资源**

- 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。

## 27.3. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 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
  ...
```

其他资源

- 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。

## 27.4. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

### 步骤

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. 保存这个文件。

6. 运行 Ansible 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 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
```

- 

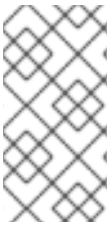
输出中没有 Inclusive Regex: initials=dp 条目确认 testing_group 自动成员规则不包含指定的条件。

其他资源

- 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。

## 27.5. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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 inventory automember-group-absent.yml
```

**其他资源**

- 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。

## 27.6. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 inventory automember-hostgroup-rule-present-copy.yml
```

**其他资源**

- 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。

## 27.7. 其他资源

- 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 28 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

## 28.1. 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。

## 28.2. 使用 IDM CLI 创建委派规则

按照以下流程，使用 IdM CLI 创建委派规则。

**先决条件**

- 您已作为 admins 组的成员登录。

**步骤**

- 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
```

## 28.3. 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。

步骤

- 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
----------------------------
```

## 28.4. 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。

步骤

- 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
```

### 28.5. 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。

**步骤**

- 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 29 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

## 29.1. 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。

## 29.2. 使用 IDM WEBUI 创建委派规则

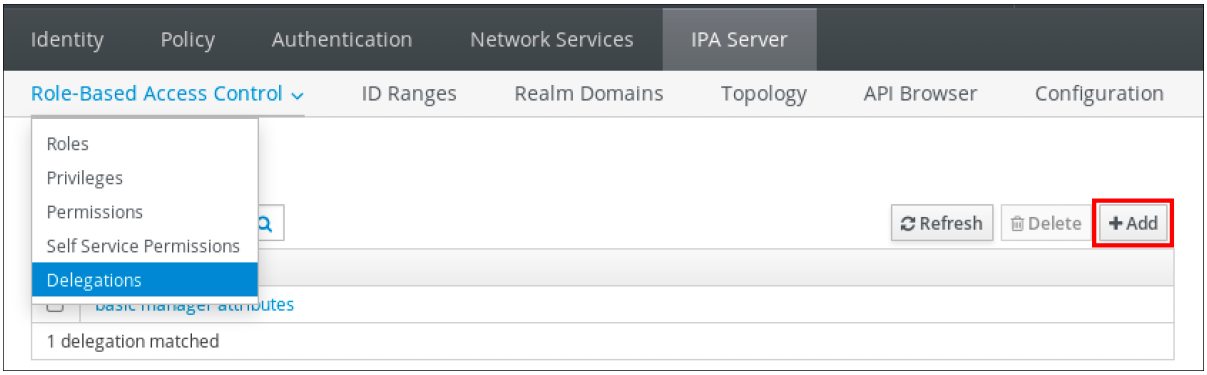
按照以下流程，使用 IdM WebUI 创建委派规则。

**先决条件**

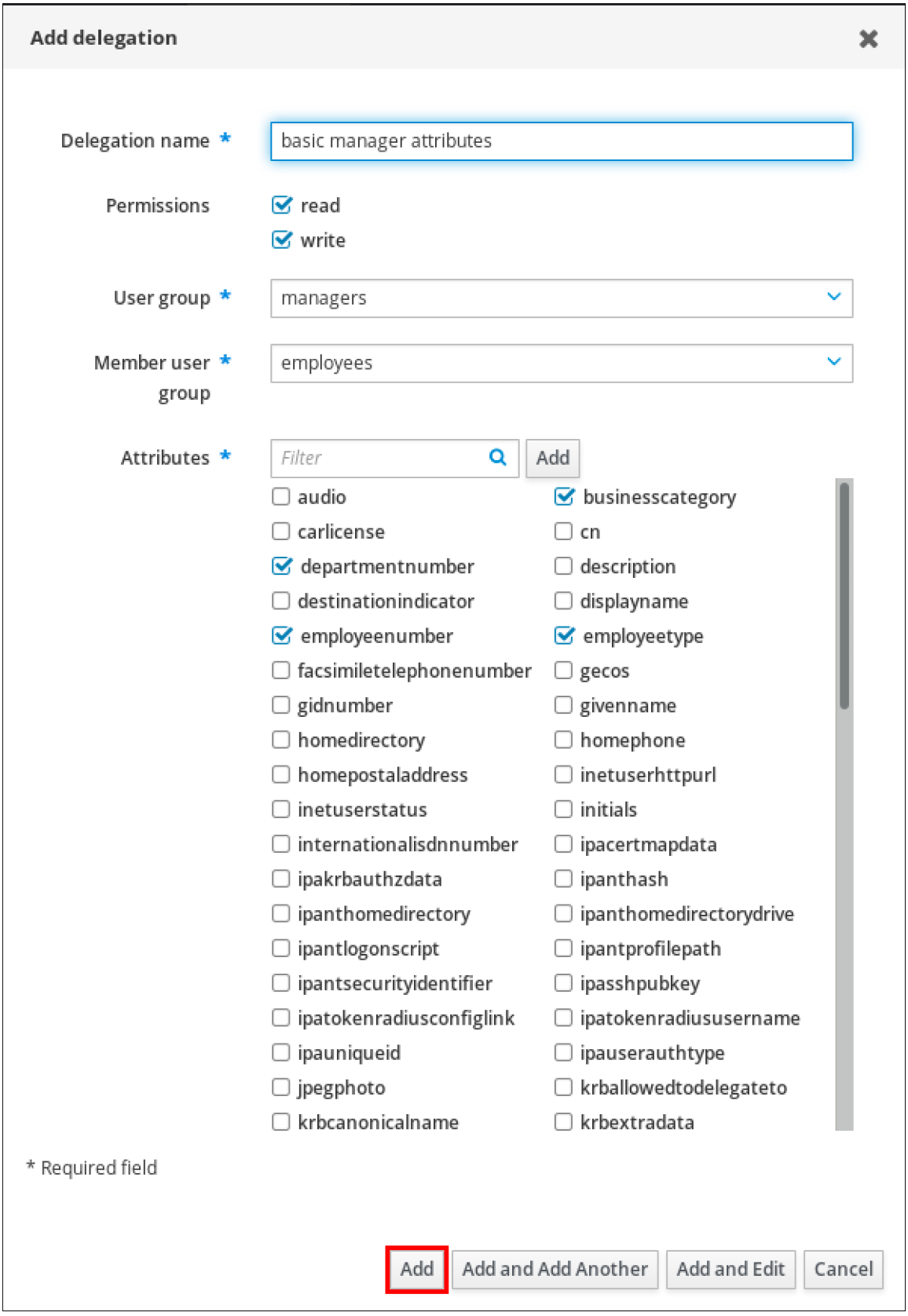
- 您已作为 admins 组的成员登录到 IdM Web UI。

**步骤**

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。

## 29.3. 使用 IDM WEBUI 查看现有的委派规则

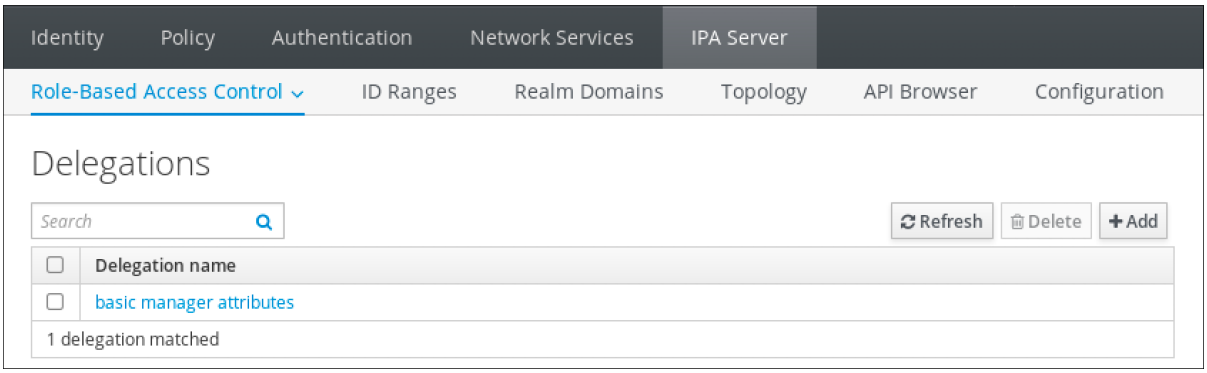
按照以下流程，使用 IdM WebUI 查看现有的委派规则。

**先决条件**

- 您已作为 admins 组的成员登录到 IdM Web UI。

**步骤**

- 在 IPA Server 菜单中点击 Role-Based Access Control → Delegations。




## 29.4. 使用 IDM WEBUI 修改委派规则

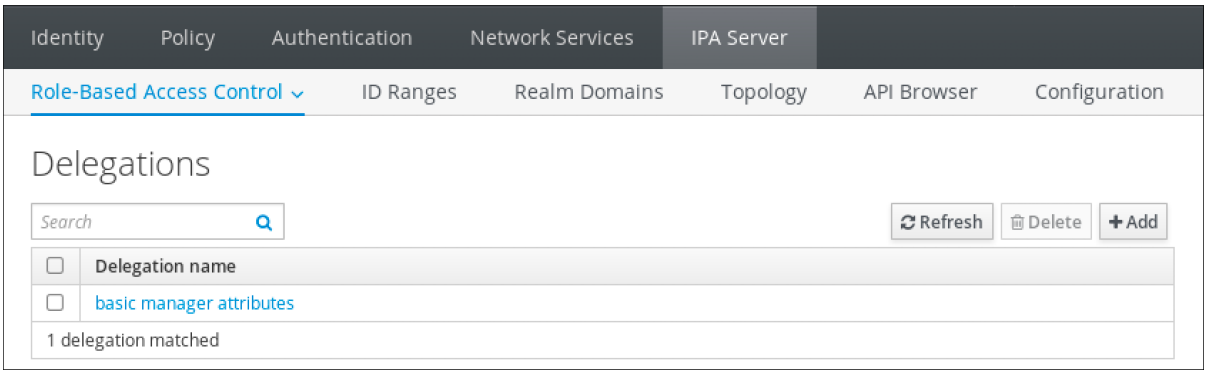
按照以下流程，使用 IdM Web UI 修改现有的委派规则。

**先决条件**

- 您已作为 admins 组的成员登录到 IdM Web UI。

**步骤**

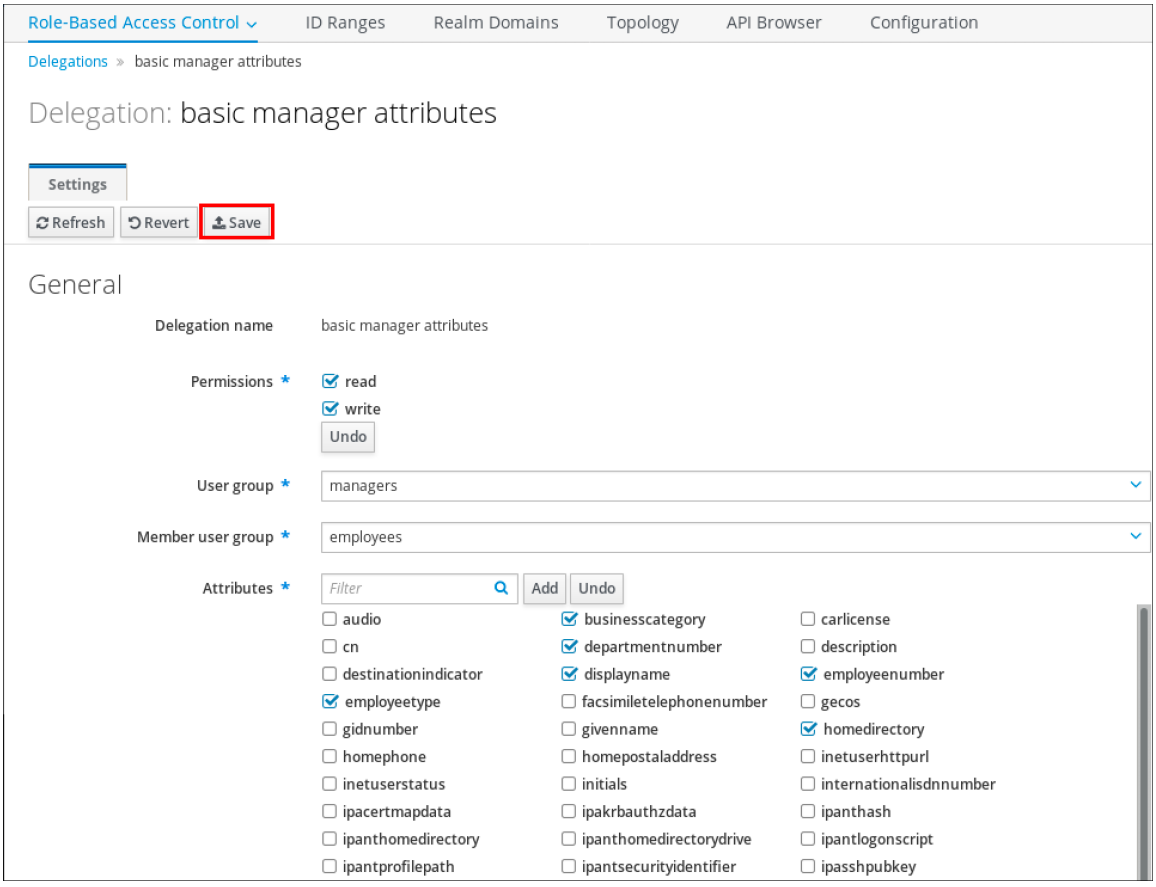
1. 在 IPA Server 菜单中点击 Role-Based Access Control → Delegations。



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。

## 29.5. 使用 IDM WEBUI 删除委派规则

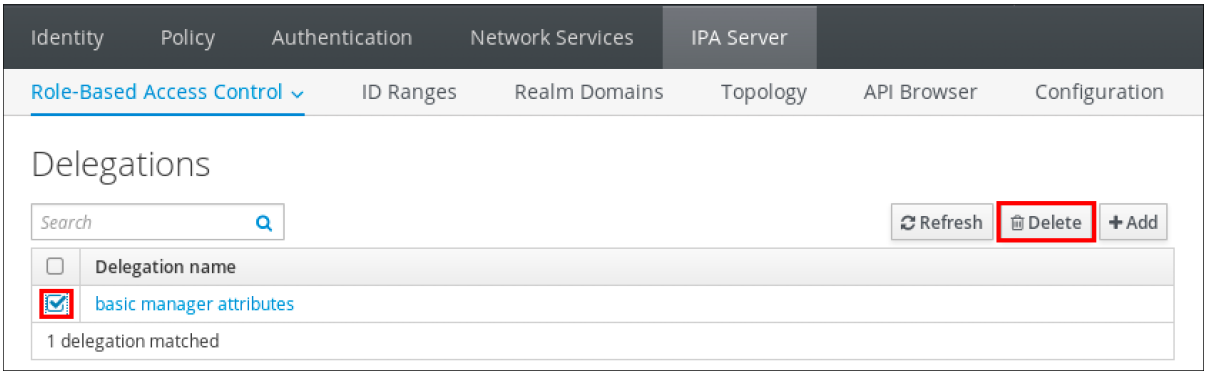
按照以下流程，使用 IdM Web UI 删除现有的委派规则。

### 先决条件

- 您已作为 **admins** 组的成员登录到 IdM Web UI。

### 步骤

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 30 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

## 30.1. 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。

## 30.2. 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。

步骤

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。

## 30.3. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 ~/MyPlaybooks/inventory
delegation-present-copy.yml
```

**其他资源**

- 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。

## 30.4. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 ~/MyPlaybooks/inventory
delegation-absent-copy.yml
```

**其他资源**

- 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。

## 30.5. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- IdM 中存在 basic manager attributes委派规则。

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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 ~/MyPlaybooks/inventory
delegation-member-present-copy.yml
```

其他资源

- 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。

## 30.6. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

- IdM 中存在 basic manager attributes委派规则。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 ~/MyPlaybooks/inventory
delegation-member-absent-copy.yml
```

**其他资源**

- 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 31 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)中基于角色的访问控制以及在命令行界面(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

## 31.1. 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、修改组并启用读访问。
- 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。

### 31.1.1. 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
- 搜索
- compare
- 添加
- 删除
- 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。

### 31.1.2. 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。

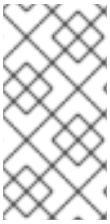
注意

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**

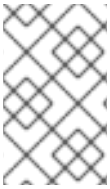
如果您试图通过命令行修改受管权限，系统不允许更改您无法修改的属性，命令会失败。如果您试图从 Web UI 修改受管权限，则无法修改的属性将被禁用。

### 31.1.3. 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。

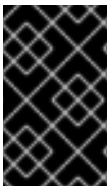


**注意**

特权可能不包含其他特权。

### 31.1.4. 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条目）的能力，特权组合更高级别任务（如在给定组中创建新用户）所需的一个或多个这些权限。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。

### 31.1.5. 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 31.1. 身份管理中的预定义角色**

| 角色 | 特权 | Description |
|---|---|---|
| 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 | Description |
| --- | --- | --- |
| 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 |

### 31.2. 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。

**步骤**

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：

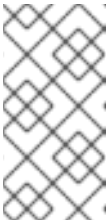
```
$ ipa permission-add "dns admin"
```

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 是更属性级的操作：您可以写入 userCertificate，而不是读取 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。

## 31.3. 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。

其他资源

- 请参阅 ipa man page。
- 请参阅 ipa help 命令。

## 31.4. 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。详情请查看链接：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。

步骤

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"
```

## 31.5. 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。

其他资源

- 请参阅 ipa man page。
- 请参阅 ipa help 命令。

### 31.6. 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。

**步骤**

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
-------------------------
```

## 31.7. 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。

其他资源

- 请参阅 ipa 手册页
- 请参阅 ipa help 命令。

# 第 32 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

## 32.1. 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、修改组并启用读访问。
- 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。

### 32.1.1. 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
- 搜索
- compare
- 添加
- 删除
- all

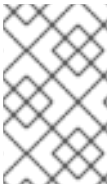
这些操作适用于三个基本目标：

- 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

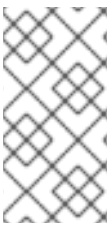
权限不能包含其他权限。

### 32.1.2. 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**

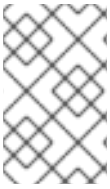
如果您试图通过命令行修改受管权限，系统不允许更改您无法修改的属性，命令会失败。如果您试图从 Web UI 修改受管权限，则无法修改的属性将被禁用。

### 32.1.3. 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。

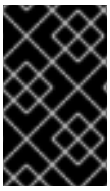


**注意**

特权可能不包含其他特权。

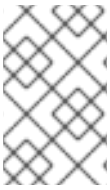
### 32.1.4. IdM 中的角色

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条目）的能力，特权组合更高级别任务（如在给定组中创建新用户）所需的一个或多个这些权限。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。

### 32.1.5. 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 32.1. 身份管理中的预定义角色**

| 角色 | 特权 | Description |
|---|---|---|
| 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 | Description |
| --- | --- | --- |
| 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 |

### 32.2. 在 IDM WEB UI 中管理权限

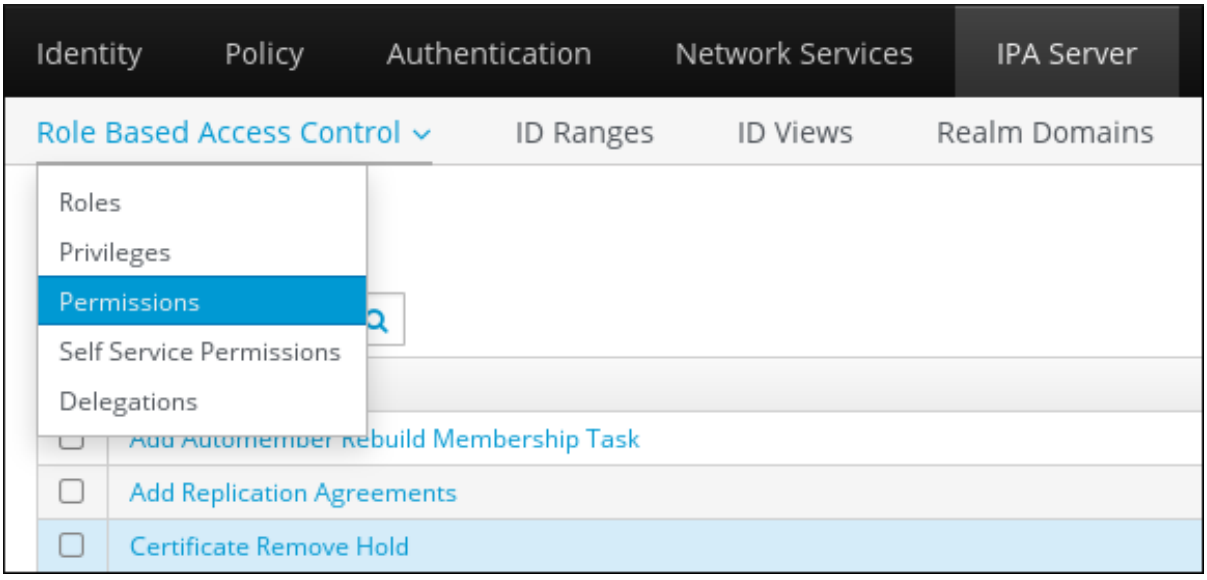
按照以下流程，使用 Web 界面(IdM Web UI)在身份管理(IdM)中管理权限。

**先决条件**

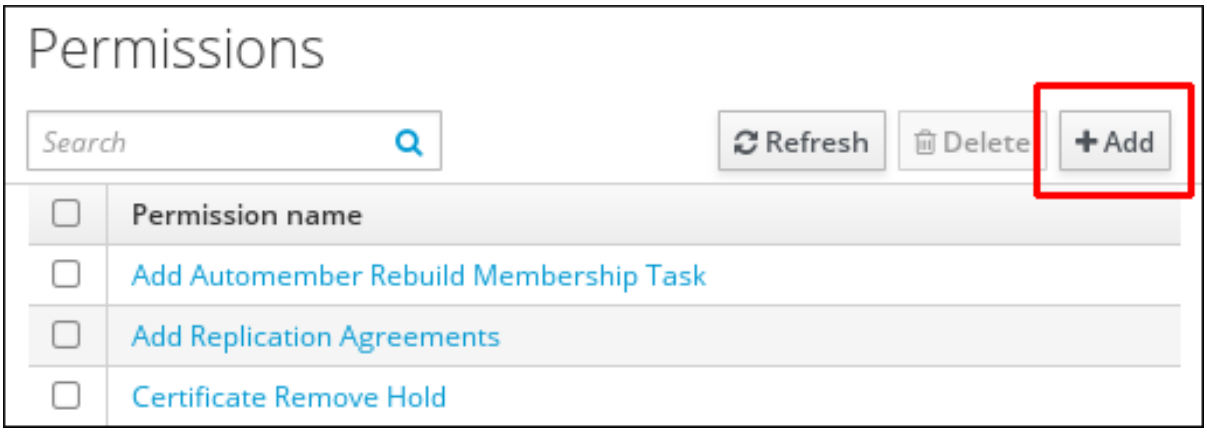
- 管理 IdM 或 用户管理员 角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

**步骤**

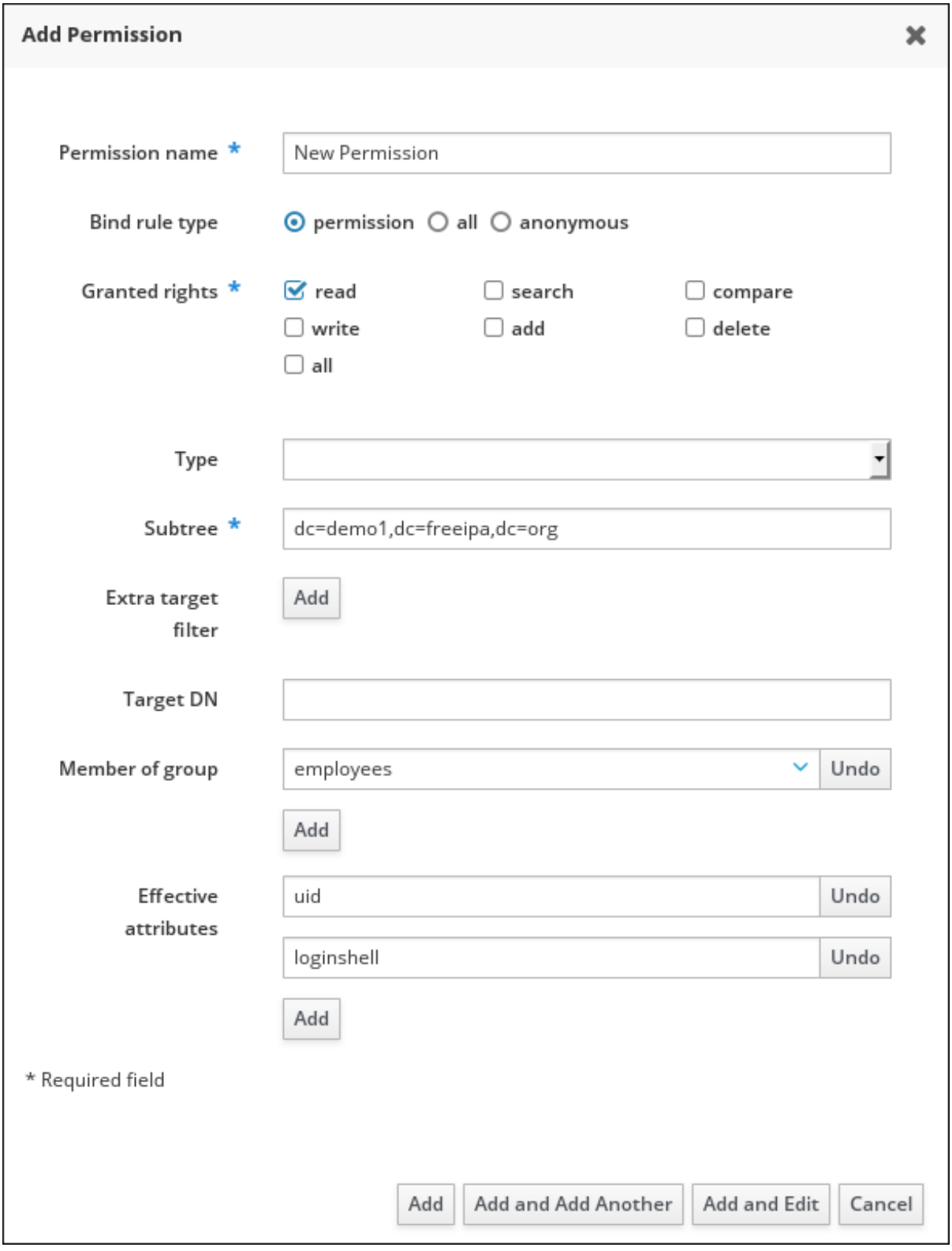
1. 要添加新权限，请在 IPA Server 选项卡中打开 Role-Based Access Control 子菜单，然后选择 Permissions ：




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。

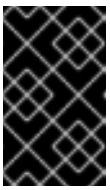
- 目标 DN 指定域名(DN)，并接受通配符。例如：uid=*,cn=users,cn=accounts,dc=com

- 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。

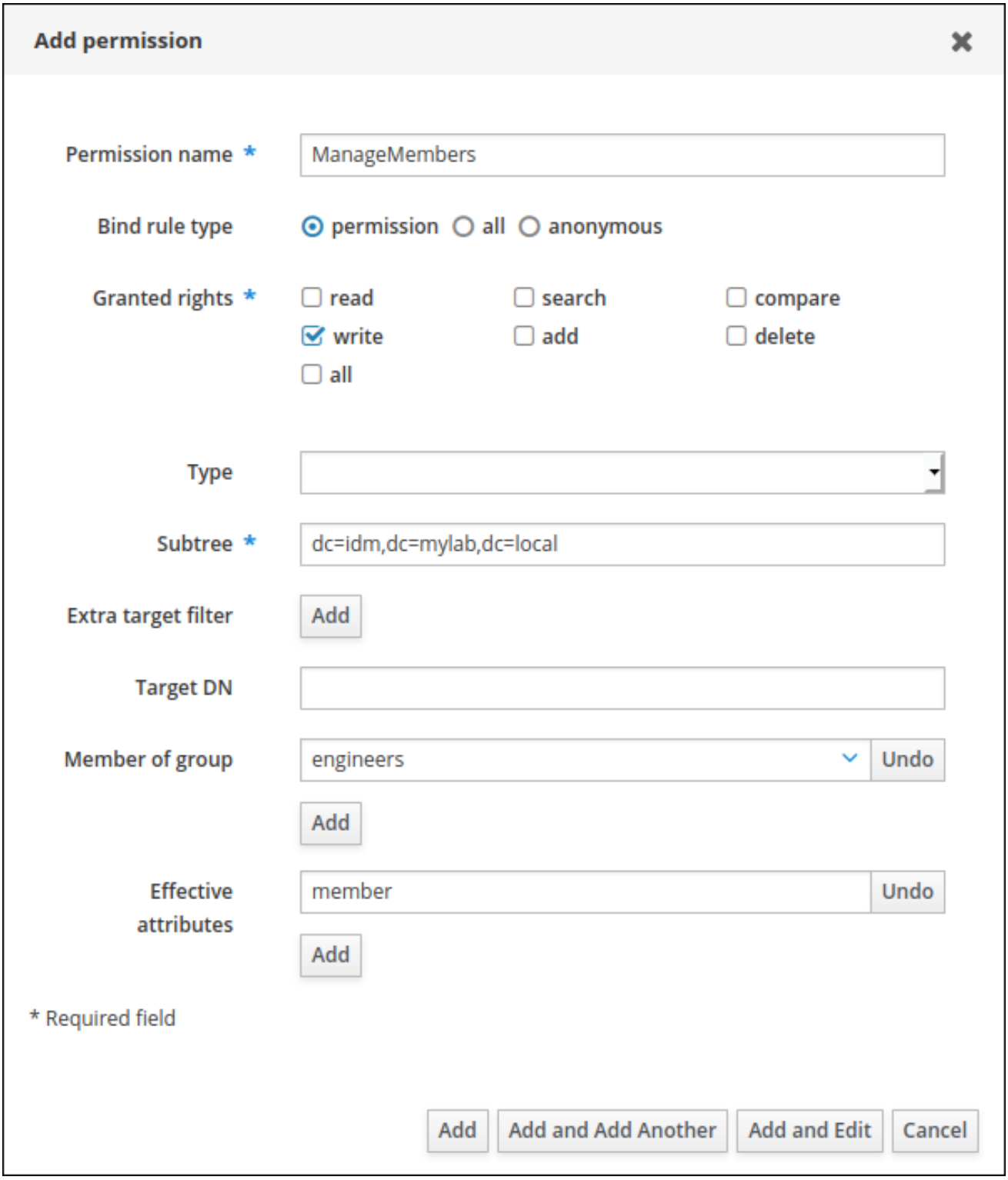
9. *可选。*您还可以通过单击权限列表中的名称来显示Permission settings 页面来编辑现有权限的属性。

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）：




### 32.3. 在 IDM WEB UI 中管理特权

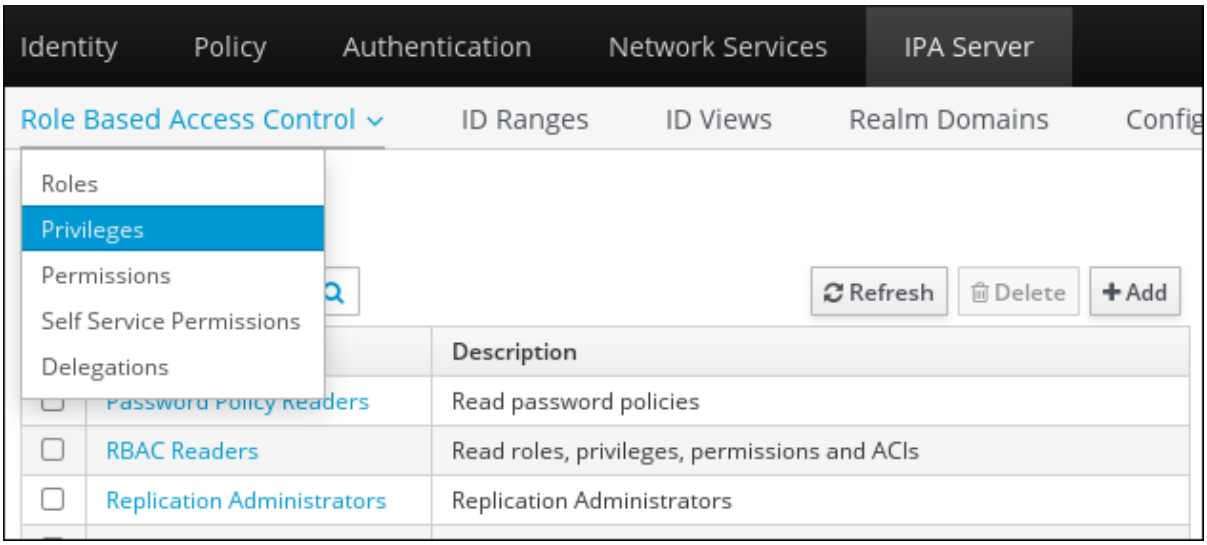
**按照以下流程，使用 Web 界面(IdM Web UI)在 IdM 中管理特权。**

**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。

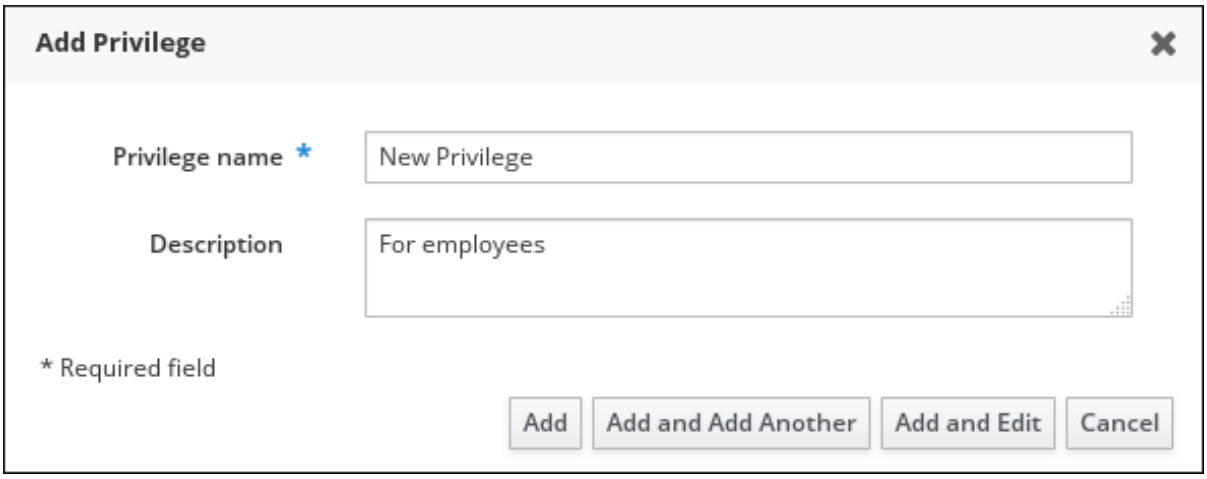
**步骤**

1. 要添加新特权，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：

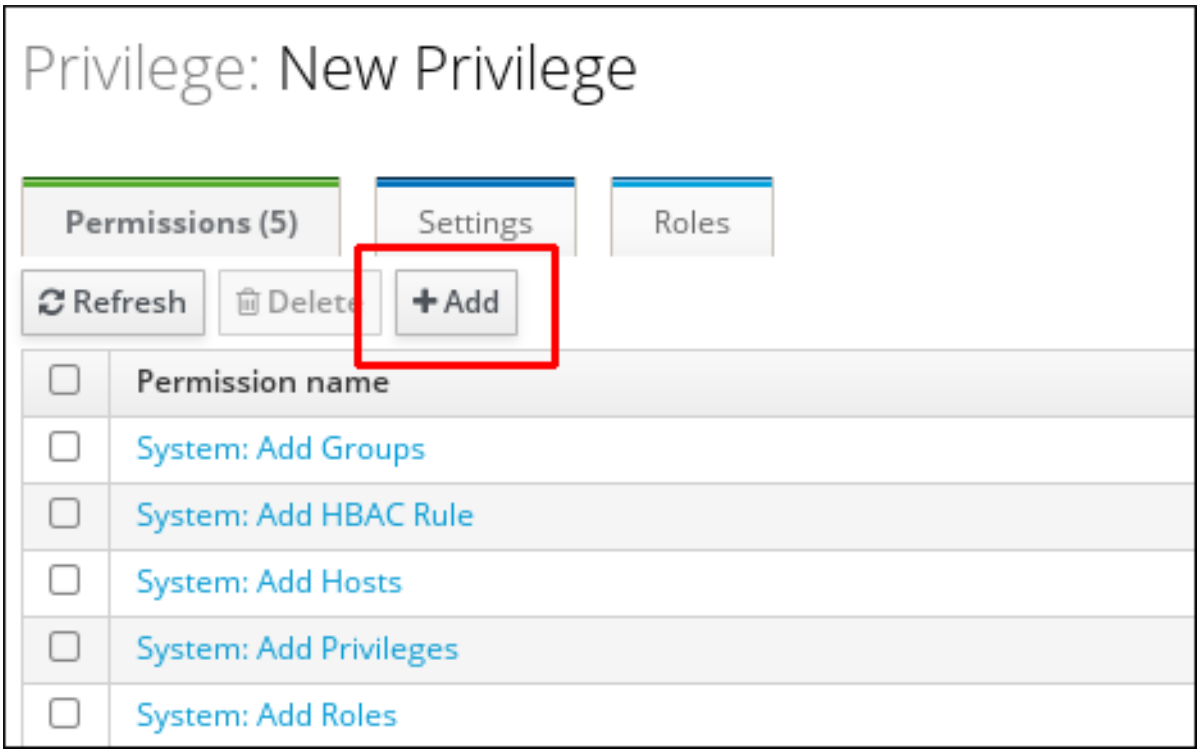
3. 此时会打开 **Add Privilege** 表单。输入特权的名称和描述：



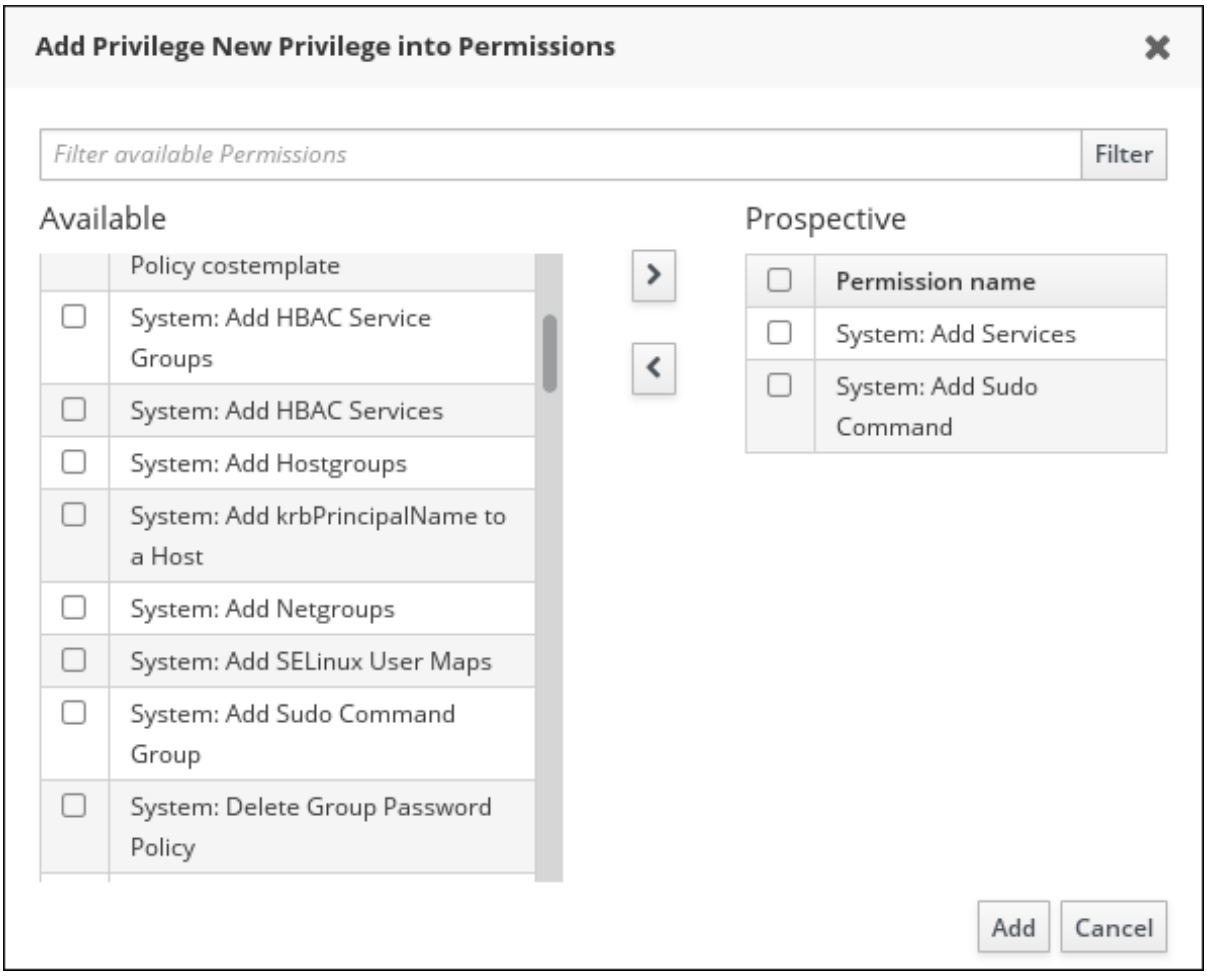

4. 单击 **Add and Edit** 按钮，以保存新特权，并继续特权配置页面来添加权限。

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。

## 32.4. 在 IDM WEB UI 中管理角色

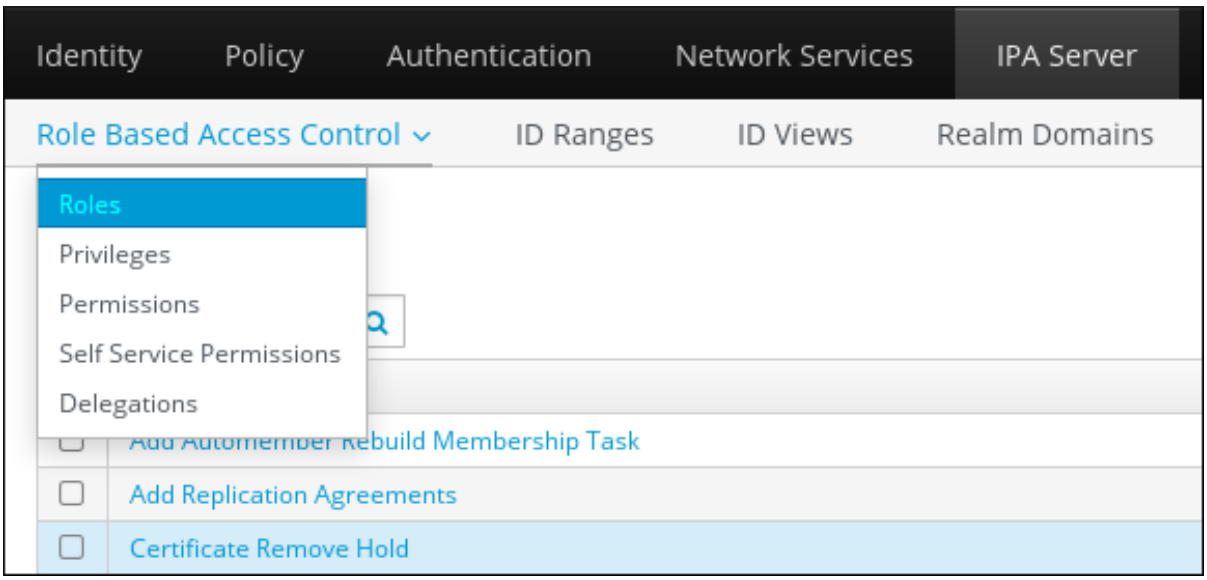
按照以下流程，使用 Web 界面(IdM Web UI)管理身份管理(IdM)中的角色。

**先决条件**

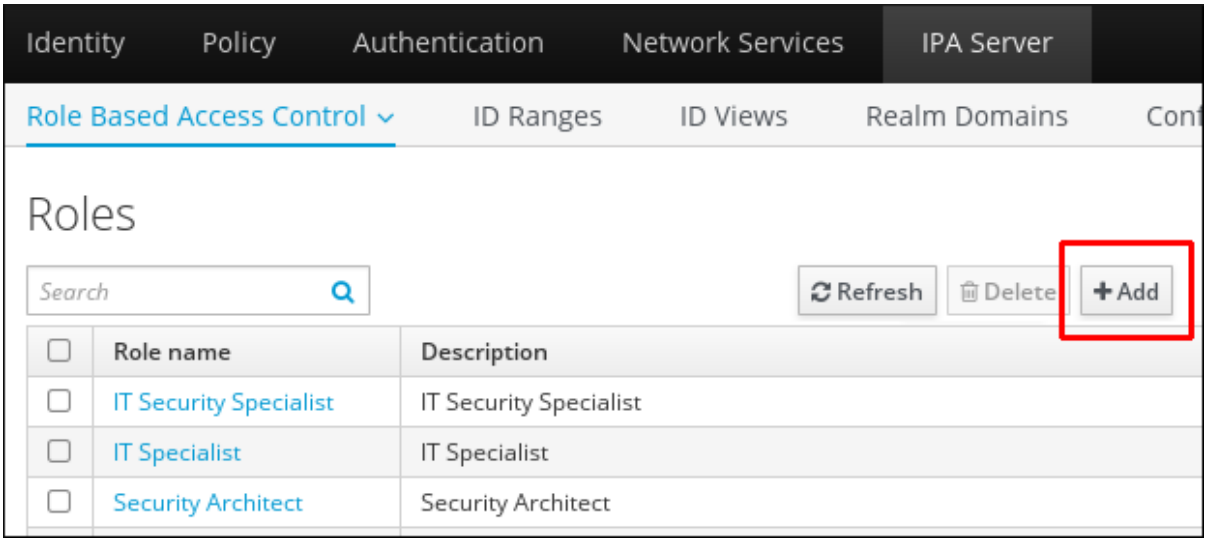
- 管理 IdM 或 用户管理员 角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。
- 现有特权。有关特权的详情，请参阅 在 IdM Web UI 中管理特权。

**步骤**

1. 要添加新角色，请在 IPA Server 选项卡中打开 Role-Based Access Control 子菜单，然后选择 Roles ：




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。

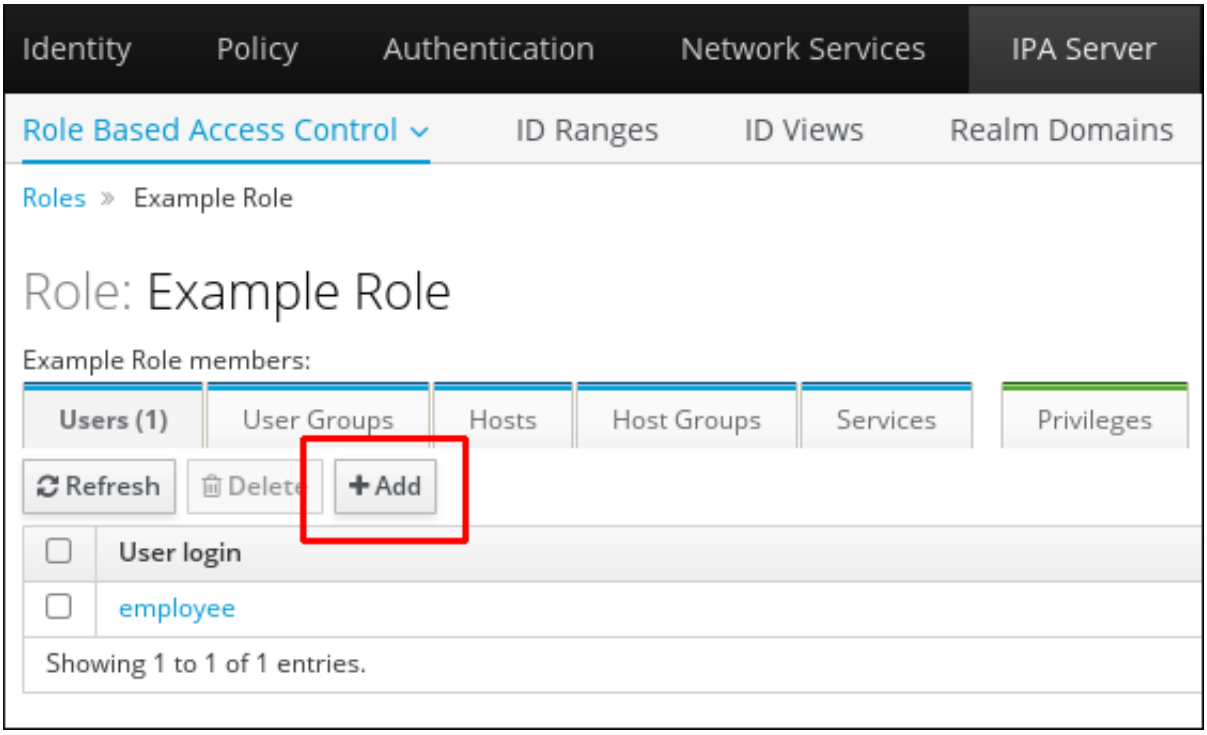



3. **此时会打开 Add Role 表单。输入角色名称和描述：**

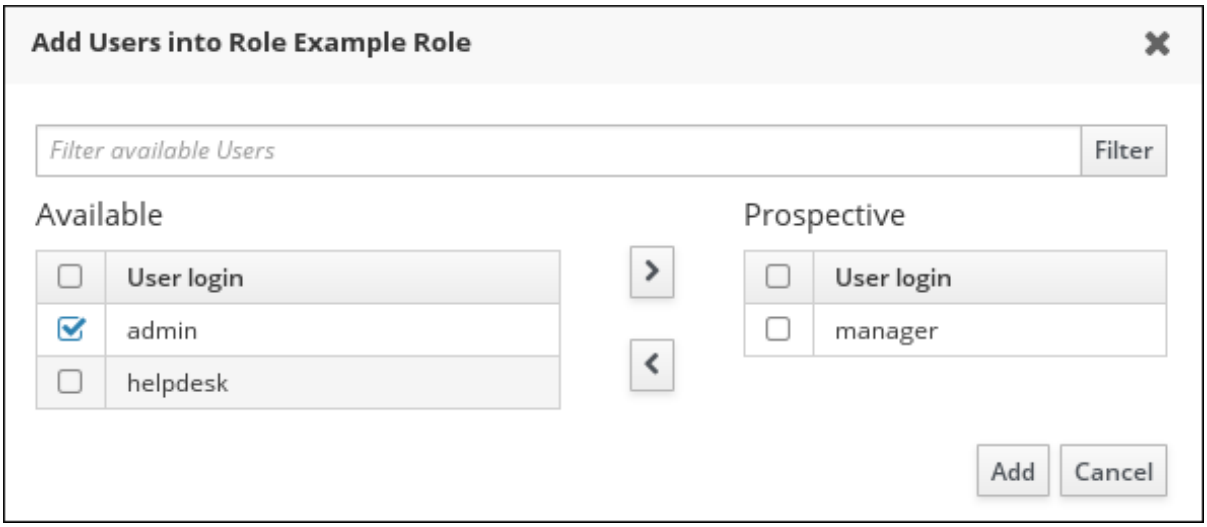

4. **单击 Add and Edit 按钮，来保存新角色，再前往角色配置页面来添加特权和用户。**

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。**

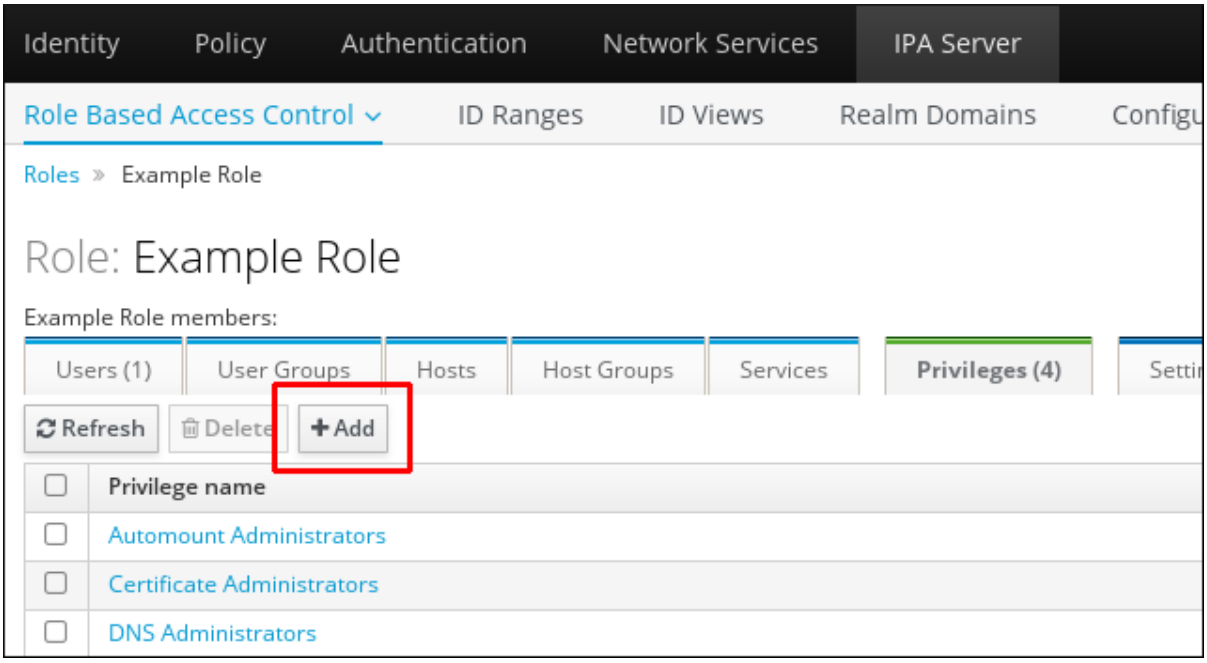
6. **单击相关列表顶部的 Add 按钮，使用 Users 、Users Groups、Hosts、Host Groups 或 Services 选项卡来添加成员。**



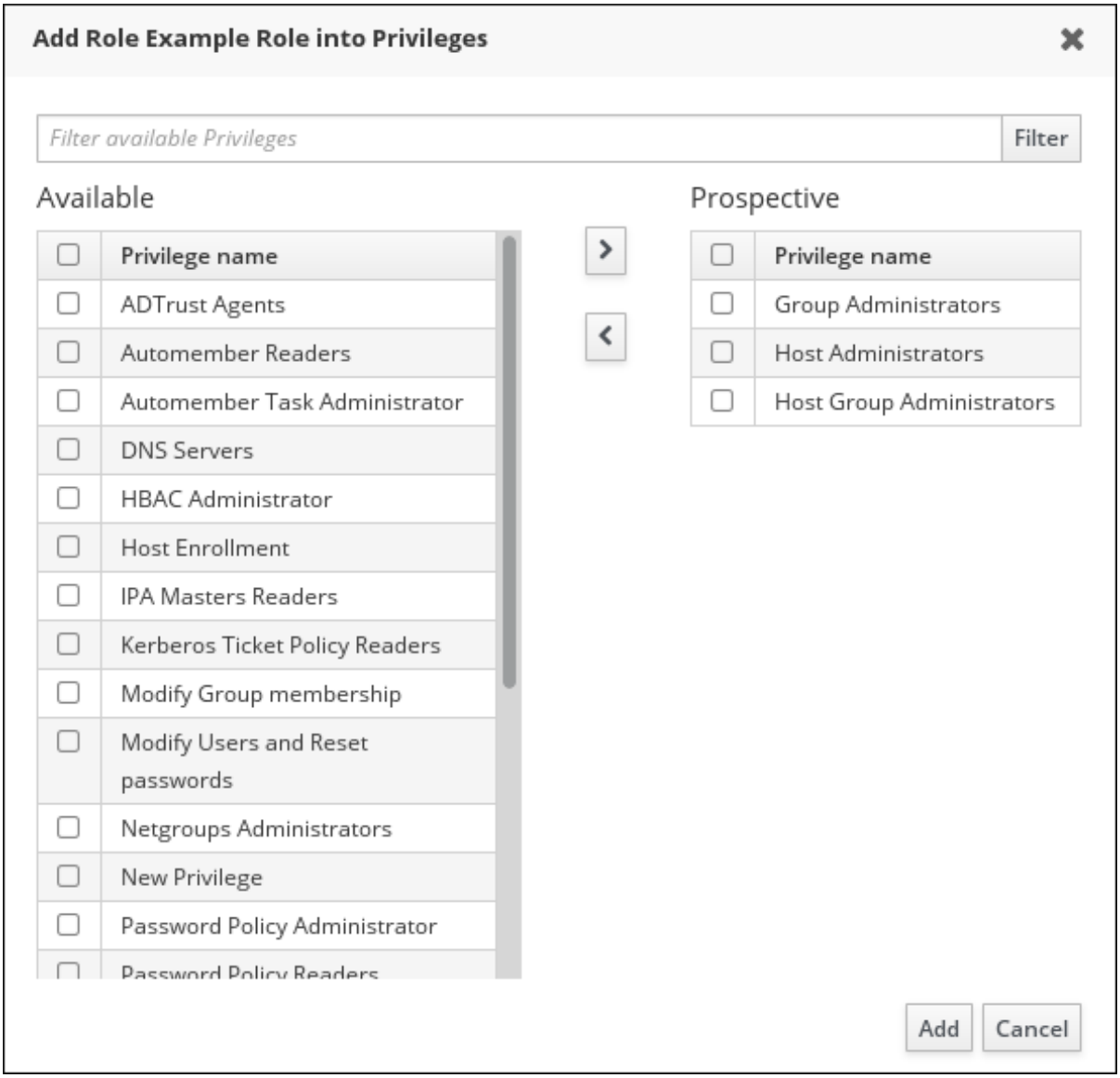

7. **在打开的窗口中，选择左侧的成员，并使用 > 按钮将它们移到 Prospective 列中。**




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。

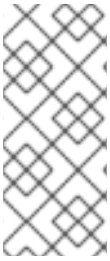
12. *可选。*如果您需要删除现有角色，请在勾选列表中其名称旁边的复选框后单击 **Delete** 按钮，来显示 **Remove roles** 对话框。

# 第 33 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，创建专用于 Ansible playbook 的子目录，如 ~/MyPlaybooks。
- 将 /usr/share/doc/ansible-freeipa/* 和 /usr/share/doc/rhel-system-roles/* 目录和子目录中的示例 Ansible playbook 复制到 ~/MyPlaybooks 目录中。
- 将清单文件包含在 ~/MyPlaybooks 目录中。

通过这个方法，您可以在一个位置找到所有 playbook，并可以在不使用 root 特权的前提下运行 playbook。



**注意**

您只需要在受管节点上具有 root 权限来执行 ipaserver、ipareplica、ipaclient 和 ipabackup 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。

**步骤**

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：

```
[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。

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
```

这些命令要求您输入 IdM admin 密码。

**其他资源**

- 请参阅 使用 Ansible playbook 安装身份管理服务器。
- 请参阅 如何构建清单。

# 第 34 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、修改组并启用读访问。
- 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

### 34.1. 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
- 搜索
- compare
- 添加
- 删除
- all

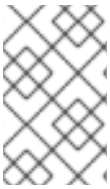
这些操作适用于三个基本目标：

- 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。

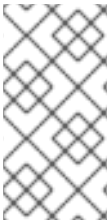
## 34.2. 默认管理的权限

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。



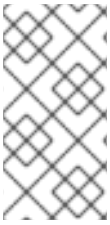
注意

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。

### 34.3. 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

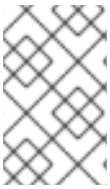
特权可能不包含其他特权。

### 34.4. 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条目）的能力，特权组合更高级别任务（如在给定组中创建新用户）所需的一个或多个这些权限。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。**

## 34.5. 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 34.1. 身份管理中的预定义角色**

| 角色 | 特权 | Description |
|---|---|---|
| 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 |

## 34.6. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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
~/<MyPlaybooks>/inventory role-member-user-present-copy.yml
```

**其他资源**

- 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。

## 34.7. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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
~/<MyPlaybooks>/inventory role-is-absent-copy.yml
```

其他资源

- 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。

### 34.8. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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
~/<MyPlaybooks>/inventory role-member-group-present-copy.yml
```

**其他资源**

- 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。

## 34.9. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

**步骤**

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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
~/<MyPlaybooks>/inventory role-member-user-absent-copy.yml
```

**其他资源**

- 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。

## 34.10. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- web_administrator 角色存在于 IdM 中。
- IdM 中存在 HTTP/client01.idm.example.com@IDM.EXAMPLE.COM 服务。

步骤

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. 保存这个文件。

6. 运行 Ansible 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
~/<MyPlaybooks>/inventory role-member-service-present-copy.yml
```

**其他资源**

- 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。

## 34.11. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

- web_administrator 角色存在于 IdM 中。
- client01.idm.example.com 主机存在于 IdM 中。

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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
~/<MyPlaybooks>/inventory role-member-host-present-copy.yml
```

**其他资源**

- 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。

## 34.12. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- web_administrator 角色存在于 IdM 中。
- web_servers 主机组存在于 IdM 中。

步骤

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. 保存这个文件。

6. 运行 Ansible 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
~/<MyPlaybooks>/inventory role-member-hostgroup-present-copy.yml
```

**其他资源**

- 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 35 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。

## 35.1. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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 inventory privilege-
present-copy.yml
```

## 35.2. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- *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。

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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 inventory privilege-member-present-copy.yml
```

## 35.3. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 inventory privilege-
member-absent-copy.yml
```

## 35.4. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- *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。

**步骤**

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. 保存这个文件。

8. 运行 Ansible 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 inventory rename-
privilege.yml
```

### 35.5. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

7. 运行 Ansible 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 inventory privilege-absent-copy.yml
```

### 35.6. 其他资源

- 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 36 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。

## 36.1. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 inventory permission-present-copy.yml
```

## 36.2. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 inventory permission-present-with-attribute.yml
```

其他资源

- 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。

## 36.3. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 inventory permission-absent-copy.yml
```

## 36.4. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- MyPermission 权限存在。

**步骤**

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. 保存这个文件。

6. 运行 Ansible 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 inventory permission-member-present-copy.yml
```

## 36.5. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

- MyPermission 权限存在。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 inventory permission-member-absent-copy.yml
```

## 36.6. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- MyPermission 存在于 IdM 中。
- IdM 中不存在 MyNewPermission。

步骤

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. 保存这个文件。

6. 运行 Ansible 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 inventory permission-renamed-copy.yml
```

## 36.7. 其他资源

- 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 37 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。

## 37.1. 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。

其他资源

- 为活动目录用户使用 ID 视图
- SSSD 客户端视图

## 37.2. 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。

## 37.3. 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)

### 37.4. 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。

步骤

- 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
[...]
```

## 37.5. 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，为特定的 IdM 客户端创建 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。

步骤

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"
```

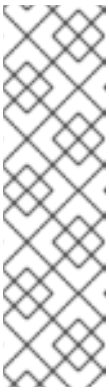
```
-------------------------
 ID View Name: example_for_host1
```

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)
```

### 37.6. 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，修改现有的 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。

**步骤**

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/
```

其他资源

- 通过修改 Default Trust View 为 AD 用户定义全局属性

## 37.7. 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，以允许用户使用 /home/user_1234/ 目录作为用

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。

**步骤**

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
```

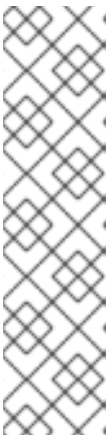
4. 将用户覆盖添加到 example_for_host1 ID 视图。覆盖用户主目录：
   - 输入 ipa idoverrideuser-add 命令
   - 添加 ID 视图的名称
   - 添加用户名，也称为锚

- 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/
```

其他资源

- 对带有 ID 视图的 IdM 客户端上的 AD 用户覆盖 Default Trust View 属性

## 37.8. 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 AD 用户的 GID，请参阅 覆盖带有 ID 视图的 IdM 客户端上 AD 用户的 Default Trust View 属性

**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）。

## 37.9. 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*/ 目录中创建了一个带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。您使用 RHEL 9.4 或更高版本。
  - 您已将 ipaadmin_password 存储在 secret.yml Ansible vault 中。

- 目标节点,也就是在其上执行 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。

**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/
```

其他资源

- ansible-freeipa 上游文档中的 idoverrideuser 模块

### 37.10. 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*/ 目录中创建了一个带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。您使用 RHEL 9.4 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,也就是在其上执行 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。

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>/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。

其他资源

- ansible-freeipa 上游文档中的 idoverrideuser 模块

## 37.11. 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 9.4 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个带有 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 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,也就是在其上执行 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。

**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、存储保护 secret.yml 文件的密码，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)
```

其他资源

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

## 37.12. 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 9.4 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个带有 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 ipaadmin_password。
- idview_for_host1 ID 视图存在。

步骤

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、存储保护 secret.yml 文件的密码，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
```

其他资源

- ansible-freeipa 上游文档中的 idoverrideuser 模块

### 37.13. 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 9.4 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个带有 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 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、存储保护 secret.yml 文件的密码，以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory ensure-idmuser-present-in-idview-with-certificates.yml
```

其他资源

- ansible-freeipa 上游文档中的 idoverrideuser 模块

## 37.14. 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 9.4 或更高版本。
- 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个带有 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 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、存储保护 secret.yml 文件的密码，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)
```

其他资源

- 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

### 37.15. 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 。

步骤

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 38 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。

## 38.1. 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 38.1. 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 38.2. 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。

**其他资源**

- 使用 ID 视图来覆盖 IdM 客户端上的用户属性值

## 38.2. 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 管理员进行了身份验证。
- 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)
```

## 38.3. 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)
```

## 38.4. 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 AD 用户的 GID，请参阅 覆盖带有 ID 视图的 IdM 客户端上 AD 用户的 Default Trust View 属性

**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 39 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。

## 39.1. 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，除了原始 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。当副本被删除时，下一个范围不会被自动分配，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。

## 39.2. 自动 ID 范围分配

### IdM ID 范围

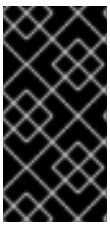
默认情况下，IdM ID 范围会在 IdM 服务器安装过程中自动分配。ipa-server-install 命令会从总共 10,000 个可能的范围中随机选择并分配 200,000 个 ID。当您决定以后合并两个独立的 IdM 域时，以这种方法选择一个随机范围可显著降低冲突 ID 的可能性。

**注意**

此 IdM ID 范围在创建后不能修改。您只能使用手动分配 DNA ID 范围 中描述的命令来手动调整分布式数字分配（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。

## 39.3. 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。

步骤

- 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。

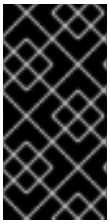
验证步骤

- 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
----------------------------
```

## 39.4. 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。您必须指定新的范围名称、范围的第一个 ID 号以及范围大小：

```
# 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 缓存并重新启动服务，SSSD 仅在更新域列表和其他存储在 IdM 服务器上的其他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
----------------------------
```

## 39.5. IDM ID 范围中的安全性和相对标识符的角色

身份管理 (IdM) ID 范围由几个参数定义：

- 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 RID 范围的第一个 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
```

安全标识符

IdM 服务器在内部使用本地域 ID 范围中的数据来为 IdM 用户和组分配唯一安全标识符 (SID)。SID 存储在用户和组对象中。用户的 SID 由以下几项组成：

- 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，并且来自此域中的用户的 RID 是 1008，则用户的 SID 为 S-1-5-21-123-456-789-1008。

相对标识符

RID 本身使用以下方法计算：

从用户的 POSIX UID 中减去范围的第一个 POSIX ID，并将相应 RID 范围的第一个 RID 添加到结果中。例如，如果 *idmuser* 的 UID 是 196600008，则第一个 POSIX ID 为 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 POSIX ID 是否属于分配的 ID 范围，然后再计算对应的 RID。例如，如果第一个 ID 是 196600000，并且范围大小为 200000，那么 POSIX ID 为 1600000，则算法不会为其计算 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* 已存在 UID 196600008，您仍然可以创建一个新的 *idmgroup* 组，GID 为 196600008。

但是，一个 SID 只能定义一个对象，一个用户 *或* 一个组。为 *idmuser* 创建的 S-1-5-21-123-456-789-1008 的 SID 无法与 *idmgroup* 共享。*必*须为 *idmgroup* 生成替代的 SID。

IdM 使用 二级相对标识符二级 RID，以避免出现冲突的 SID。这个二级 RID 由以下内容组成：

- 二级 RID 基础

- 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 基础被设置为 1000000。要计算新创建的 *idmgroup* 的 RID：减少用户 POSIX UID 范围内的第一个 POSIX ID，并将二级 RID 范围的第一个 RID 添加到结果中。因此，*idmgroup* 被分配 1000008 的 RID。因此，*idmgroup* 的 SID 是 S-1-5-21-123-456-789-1000008。

只有当之前使用手动设置 POSIX ID 创建用户或组对象时，IdM 使用二级 RID 来计算 SID。否则，自动分配会防止分配相同的 ID 两次。

**其他资源**

- 使用 Ansible 添加新的本地 IdM ID 范围

## 39.6. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

**步骤**

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、存储保护 secret.yml 文件的密码，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 缓存并重新启动服务，SSSD 仅在更新域列表和其他存储在 IdM 服务器上的其他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
----------------------------
```

- 

其他资源

- IdM ID 范围中的安全性和相对标识符的角色

## 39.7. 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。

步骤

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
```

其他资源

- 请参阅 使用命令行删除信任。
- 请参阅 使用 IdM Web UI 删除信任。

## 39.8. 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)。

步骤

- 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
```

## 39.9. 手动 ID 范围分配

在某些情况下，需要手动分配分布式数字分配（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。

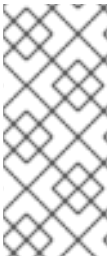
    有关如何创建新 IdM ID 范围的详情，请参考 添加新的 IdM ID 范围。

- 副本停止工作

  当副本停止正常工作且必须被删除时，不会自动检索副本的 DNA ID 范围，这意味着之前分配给副本的 DNA ID 范围变得不可用。您要恢复 DNA ID 范围，并使其可用于其他副本。

  为此，请在手动将该范围分配给其他服务器之前找出 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 中的命令来手动分配 DNA ID 。



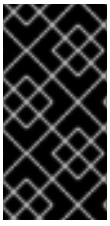
注意

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。

## 39.10. 手动分配 DNA ID 范围

在某些情况下，您可能需要为现有副本手动分配分布式 数字分配（DNA）ID 范围，例如将分配给不工作副本的 DNA ID 范围重新分配。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-manage dnarange-set ：

```
# ipa-replica-manage dnarange-set serverA.example.com 1250-1499
```

- 要为指定服务器定义下一个 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 40 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。

## 40.1. 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。
- 您已获得 IdM admin 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。
- 您有访问您要执行流程的 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：
   - 生成并将 subID 范围分配给一个 IdM 用户：

```
# 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
```

- 生成并将 subID 范围分配给所有 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了 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
```

## 40.2. 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。

### 40.3. 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
```

## 40.4. 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。
- shadow-utils-subid 软件包已安装。
- 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 41 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。

## 41.1. 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。

## 41.2. 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，并允许它识别它将要连接的 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。

## 41.3. 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。

## 41.4. 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存在许多相似性，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 41.1. 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 41.2. 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。

## 41.5. 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，以及执行这些操作的先决条件、上下文和后果。

**表 41.3. 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 41.4. 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>**)。 |

### 41.6. 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的信息。**

**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 41.5. 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。 |

## 41.7. 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 41.1. 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 41.2. 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。

## 41.8. 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
```

## 41.9. 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。

### 41.9.1. 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

### 41.9.2. 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:
```

**其他资源**

- 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。

### 41.9.3. 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。

**步骤**

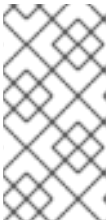
按照以下流程，使用客户端系统的 keytab 以非交互方式重新注册身份管理(IdM)客户端。例如，使用客户端 keytab 重新注册适用于自动安装。

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。

### 41.9.4. 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 ~]$
```

## 41.10. 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。

### 41.10.1. 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
```

### 41.10.2. 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。

**步骤**

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
```

### 41.10.3. 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。

### 41.10.4. 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. 为在 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。

## 41.11. 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。

### 41.11.1. 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。

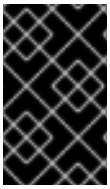
禁用主机可防止域用户访问它，而不必将其从域中永久删除。

步骤

- 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。

### 41.11.2. 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。

步骤

- 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 42 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。

## 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，并允许它识别它将要连接的 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存在许多相似性，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：

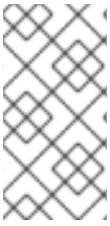
- SSH 密钥。主机的 SSH 公钥已创建并上传到主机条目。从那里，系统安全服务守护进程 (SSSD) 使用 IdM 作为身份提供程序，并可与 OpenSSH 和其他服务一起引用位于 IdM 中的公钥。
- 计算机证书。在这种情况下，计算机使用由 IdM 服务器的证书认证机构签发的 SSL 证书，然后存储在 IdM 的目录服务器中。证书然后发送到计算机，当它向服务器进行身份验证时会存在该证书。在客户端上，证书由名为 certmonger 的服务管理。

## 42.5. 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的信息。

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 42.3. 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。 |

### 42.6. 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。**

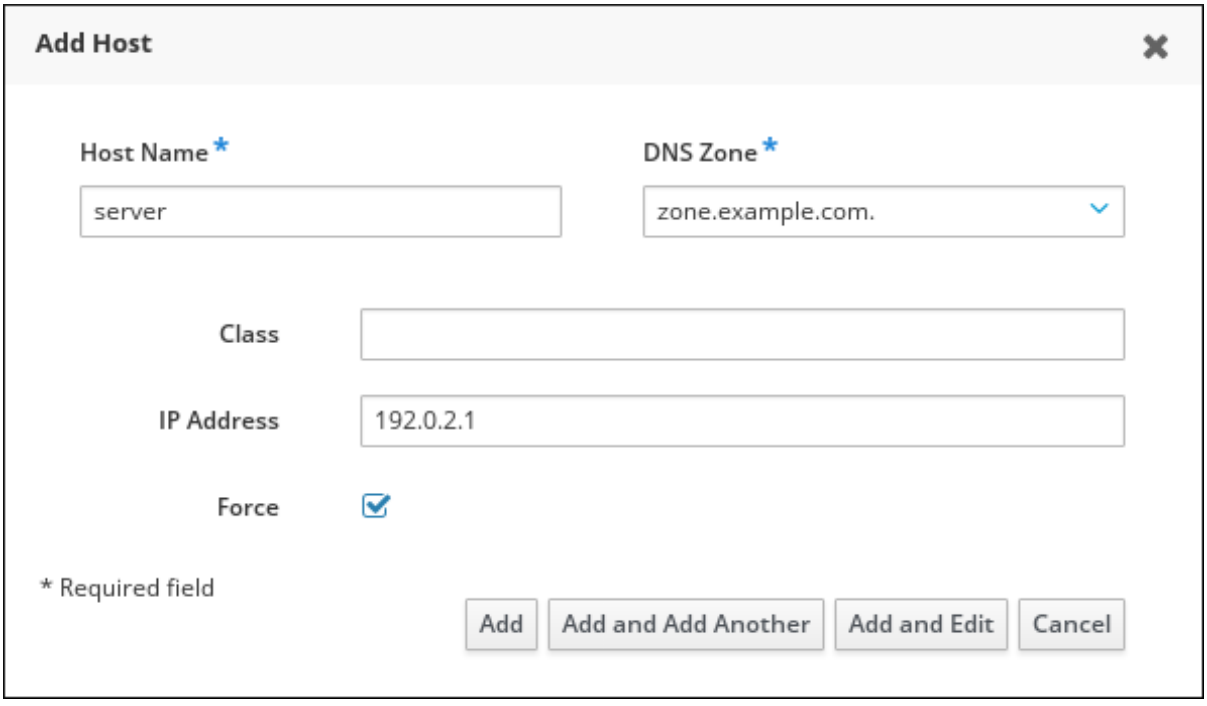
2. 单击主机列表顶部的 **Add**。

**图 42.1. 添加主机条目**

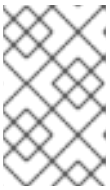

3. 输入机器名称，并在下拉列表中配置的区中选择域。如果已经为主机分配了静态 IP 地址，则将它与主机条目一起包含，以便完全创建 DNS 条目。

Class 字段目前没有特定的目的。

**图 42.2. 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。

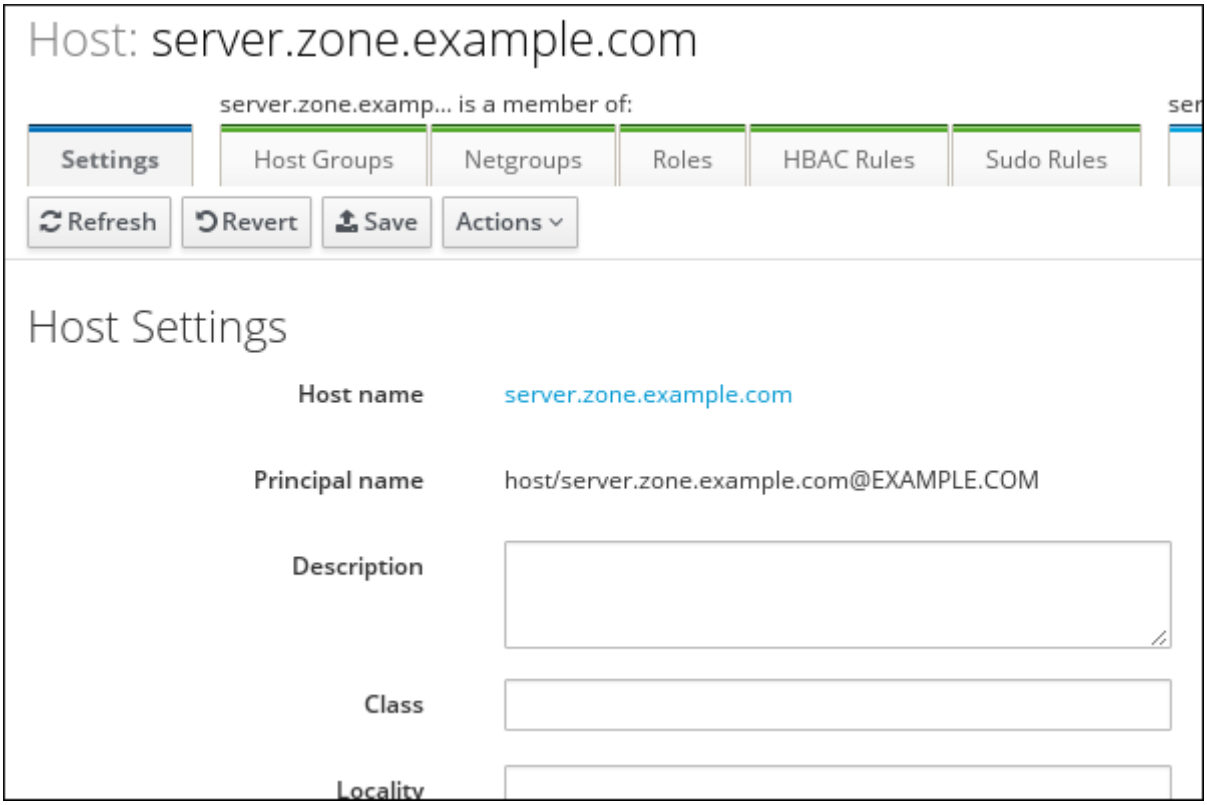


**注意**

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 42.3. 扩展的条目页面**



# 第 43 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，要执行以下概念和操作：

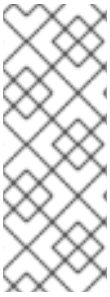
- 确保存在的 IdM 主机条目仅由 FQDN定义
- 确保存在带有 IP 地址的 IdM 主机条目
- 确保存在带有随机密码的多个 IdM 主机条目
- 确保存在带有多个 IP 地址的 IdM 主机条目
- 确保 IdM 主机条目不存在

## 43.1. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

**步骤**

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。

## 43.2. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

**步骤**

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
```

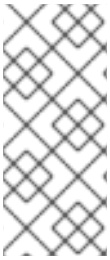
2. 创建一个 Ansible playbook 文件，其中包含您要确保的 IdM 中的 完全限定域名 (FQDN)。另外，如果 IdM 服务器配置为管理 DNS，并且您知道主机的 IP 地址，请为 ip_address 参数指定一个值。主机需要 IP 地址才能存在于 DNS 资源记录中。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-freeipa/playbooks/host/host-present.yml 文件中的示例。您还可以包含其他附加信息：

```
---
- 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。

## 43.3. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

步骤

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**

要使用随机的、一次性密码(OTP)来将主机部署为 IdM 客户端，请参阅 使用 Ansible playbook 进行 IdM 客户端注册的授权选项 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:
```

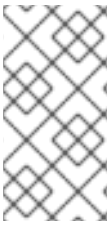
2. 输入 ipa host-show 命令并指定其中一个主机的名称：

```
$ 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。

## 43.4. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

**步骤**

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，在单独的行上指定多个 IPv4 和 IPv6 *ip_address* 值。要简化此步骤，您可以复制并修改 /usr/share/doc/ansible-

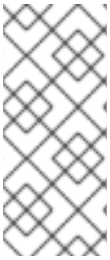
freeipa/playbooks/host/host-member-ipaddresses-present.yml 文件中的示例。您还可以包含附加信息：

```
---
- 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。

## 43.5. 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

### 步骤

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。

### 43.6. 其他资源

- 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 44 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

## 44.1. 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。

## 44.2. 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。

步骤

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
----------------------------
```

## 44.3. 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。

**步骤**

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
---------------------
```

## 44.4. 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。

步骤

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。

### 44.5. 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。

步骤

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。

### 44.6. 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。

**步骤**

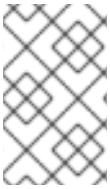
1. 要删除主机组成员，请使用 ipa hostgroup-remove-member 命令，并提供相关信息。您可以使用这些选项指定要删除的成员类型：

   - 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}
```

## 44.7. 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。

**步骤**

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：

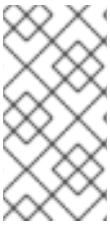
   ```
   $ ipa hostgroup-add-member-manager group_name --user example_member
   Host-group: group_name
   Member hosts: server.idm.example.com
   Member host-groups: project_admins
   Member of netgroups: group_name
   Membership managed by users: example_member
   -------------------------
   Number of 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
```

其他资源

- 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。

### 44.8. 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。

步骤

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。

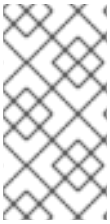
   例如，要从名为 *group_name* 的组中删除作为成员管理者的名为 *example_member* 的用户：

```
$ 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
```

其他资源

- 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 45 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

## 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. 在 IDM WEB UI 中查看主机组

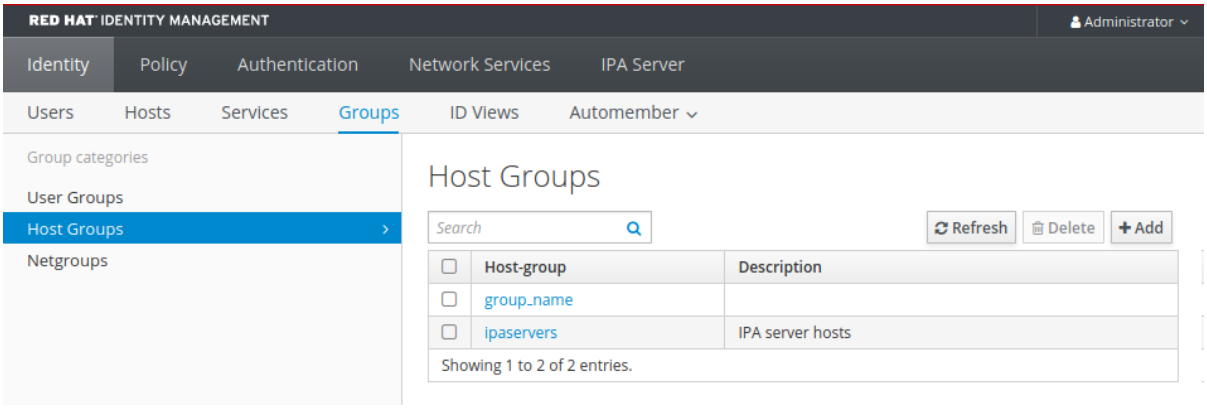
按照以下流程，使用 Web 界面(Web UI)查看 IdM 主机组。

先决条件

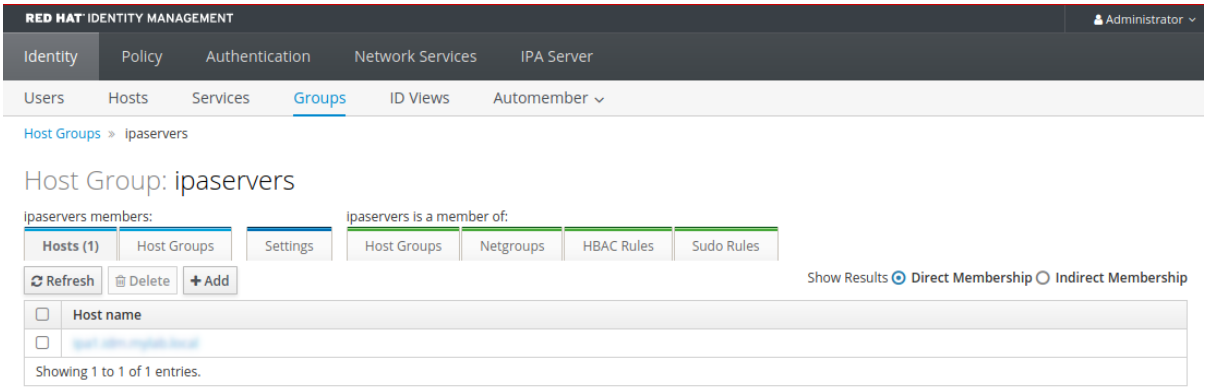
- 管理 IdM 或用户管理员角色的管理员特权。
- 您已登录到 IdM Web UI。详情请参阅 在 Web 浏览器中访问 IdM Web UI。

步骤

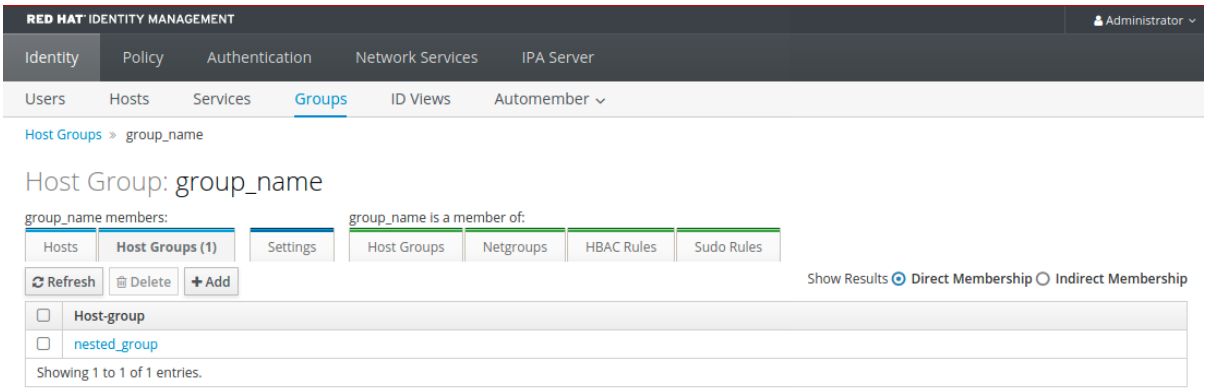
1. 点击 Identity → Groups，然后选择 Host Groups 选项卡。
   - 页面中列出了现有的主机组及其描述。
   - 您可以搜索特定的主机组。




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。**




## 45.3. 在 IDM WEB UI 中创建主机组

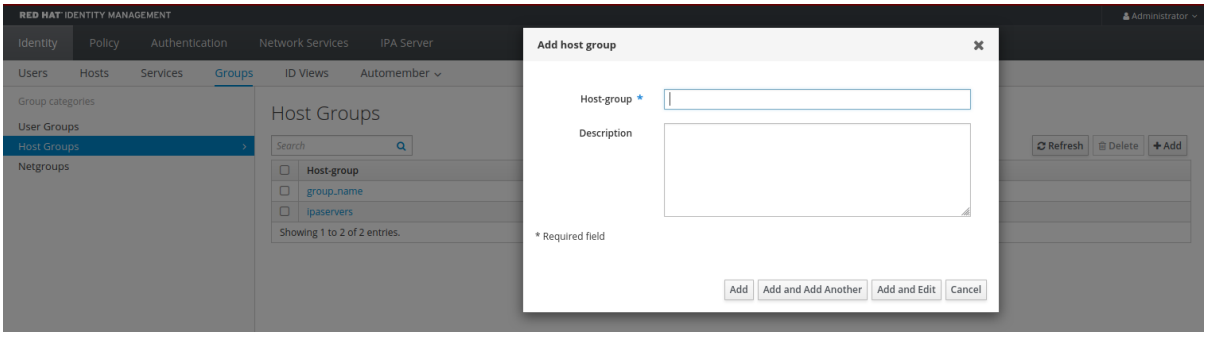
按照以下流程，使用 Web 界面(Web UI)创建 IdM 主机组。

**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。

**步骤**

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。




## 45.4. 在 IDM WEB UI 中删除主机组

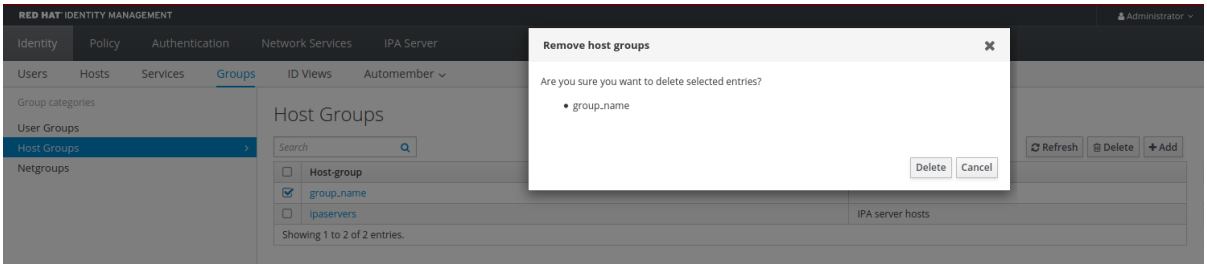
按照以下流程，使用 Web 界面(Web UI)删除 IdM 主机组。

**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。

步骤

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。

## 45.5. 在 IDM WEB UI 中添加主机组成员

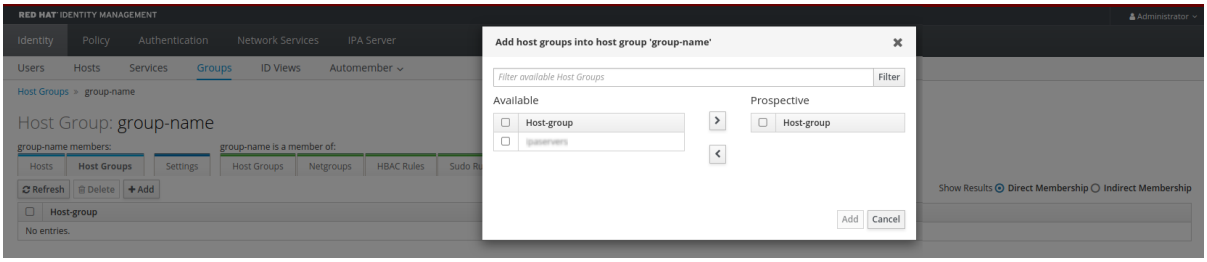
按照以下流程，使用 Web 界面(Web UI)在 IdM 中添加主机组成员。

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。

**步骤**

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。




## 45.6. 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。

**步骤**

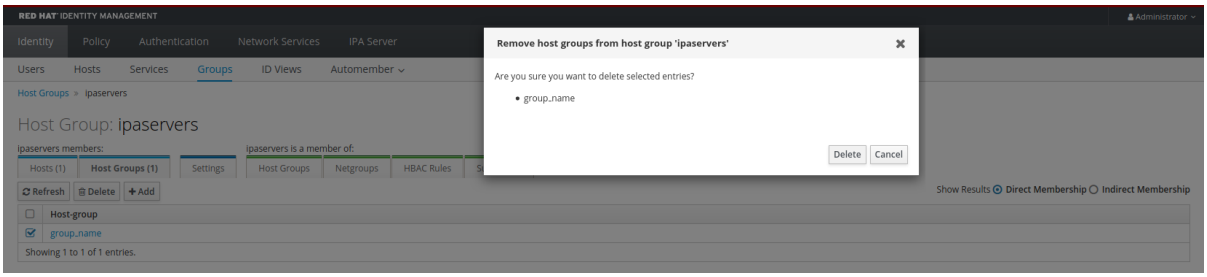
1. 点击 **Identity → Groups**，并选择 **Host Groups** 选项卡。

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。

## 45.7. 使用 WEB UI 添加 IDM 主机组成员管理者

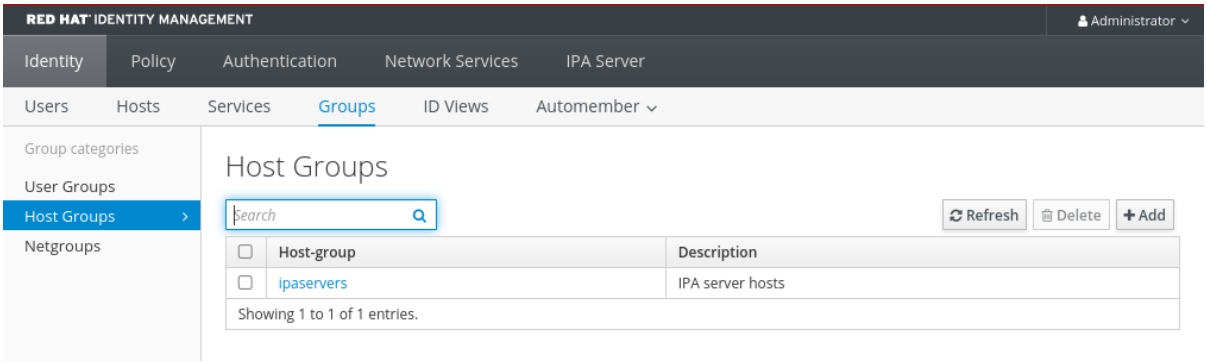
按照以下流程，使用 Web 界面(Web UI)将用户或用户组作为主机组成员管理者添加到 IdM 中。成员管理者可以将主机组成员管理者添加到 IdM 主机组中，但不能更改主机组的属性。

**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。

**步骤**

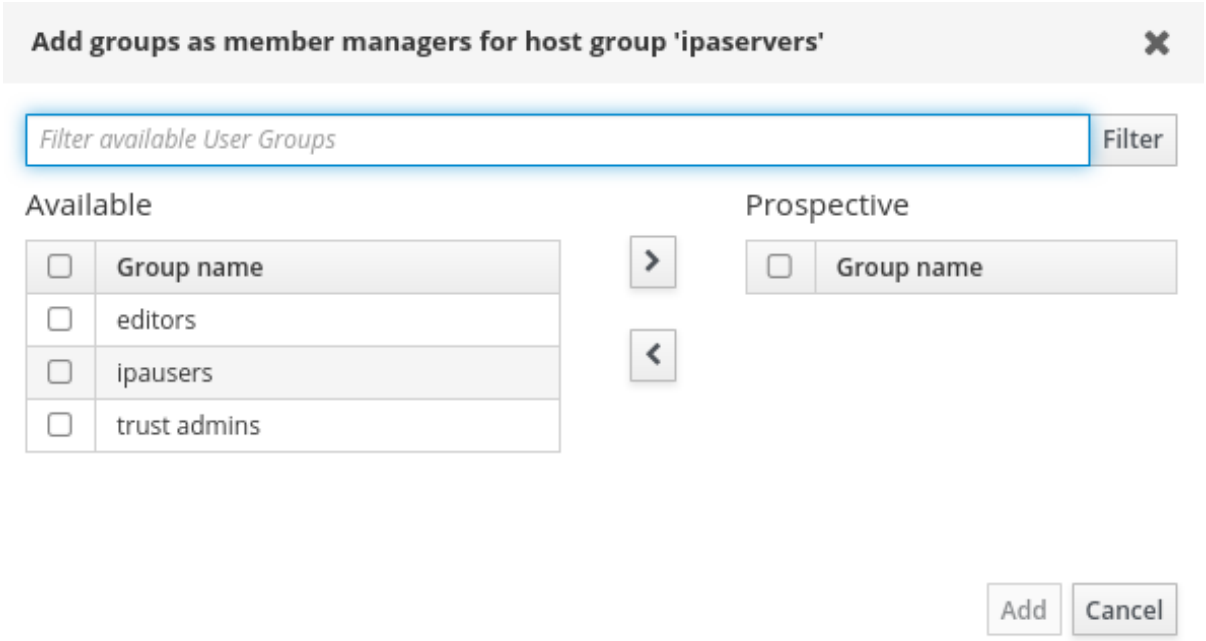
1. 点击 **Identity → Groups，并选择 Host Groups** 选项卡。




2. 单击要添加成员管理者的组的名称。

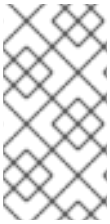
3. 单击 **member managers** 选项卡 **User Groups or Users**，具体取决于您要添加的成员管理者的类型。此时会出现相应的对话框。

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。

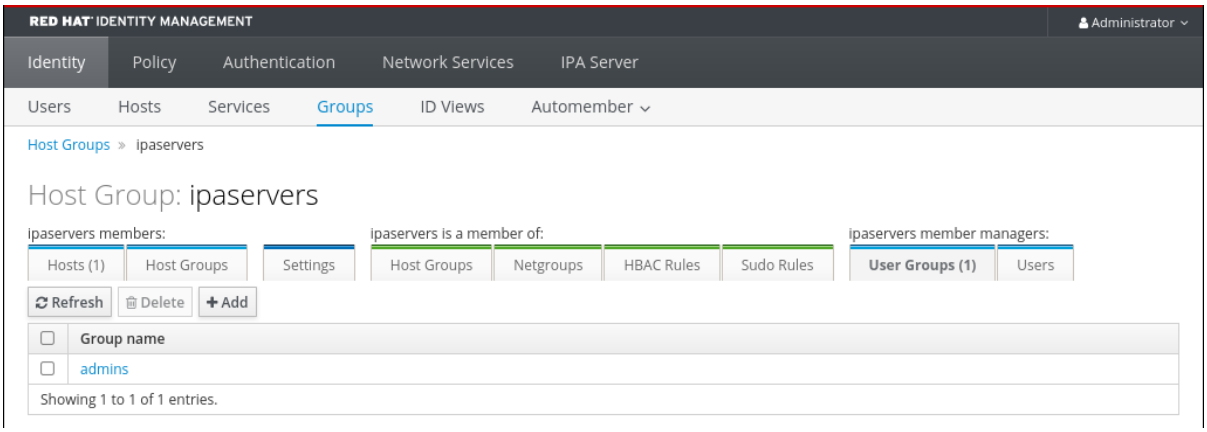


**注意**

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。




## 45.8. 使用 WEB UI 删除 IDM 主机组成员管理者

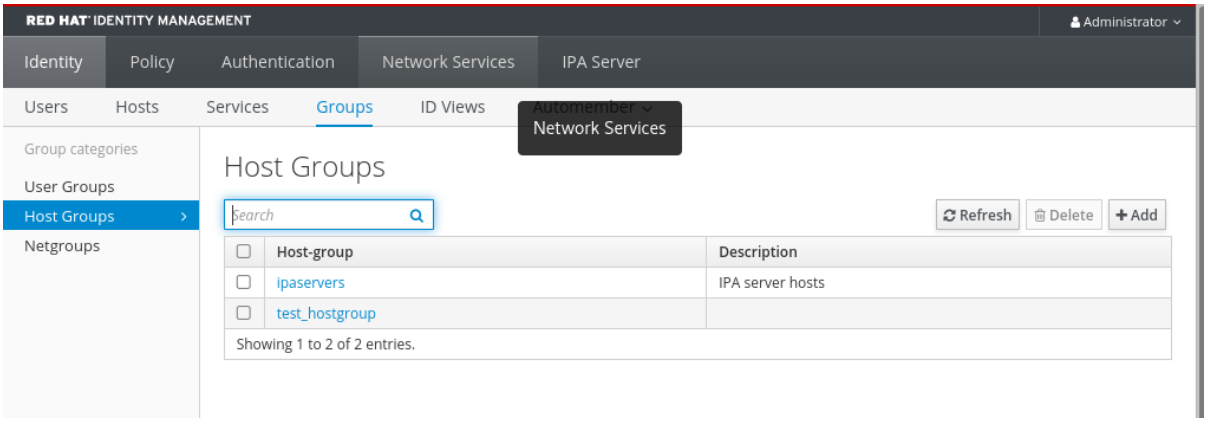
按照以下流程，使用 Web 界面(Web UI)在 IdM 中删除作为主机组成员管理者的用户或用户组。成员管理者可以从 IdM 主机组中删除主机组成员管理者，但不能更改主机组的属性。

**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。

**步骤**

1. 点击 **Identity** → **Groups**，并选择 **Host Groups** 选项卡。

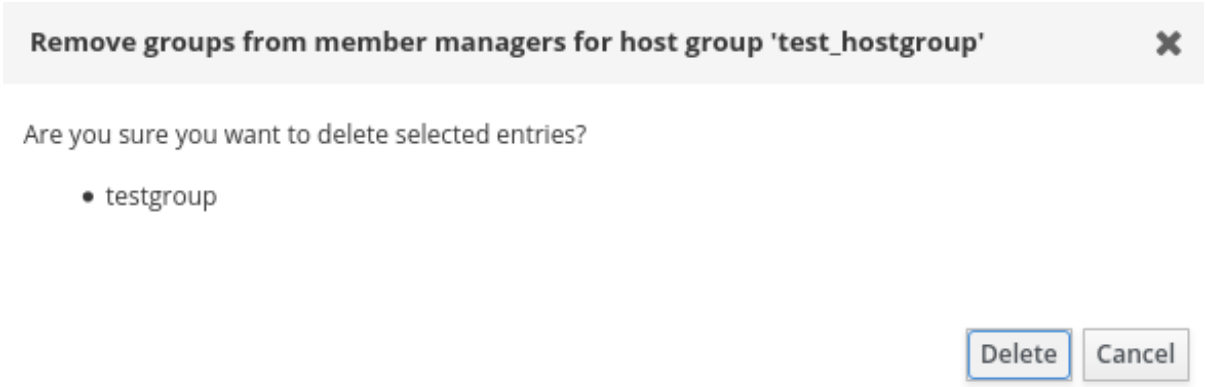



2. 单击您要从中删除成员管理者的组的名称。

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。

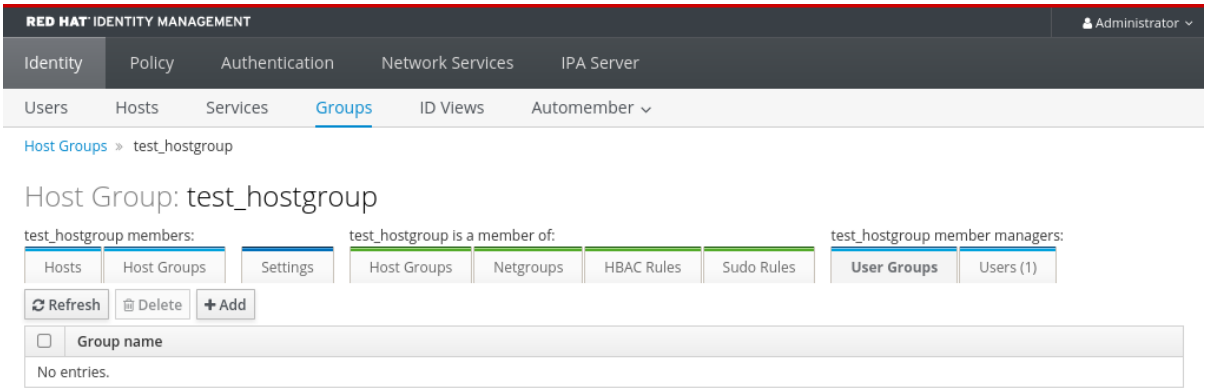



**注意**

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 46 章 使用 ANSIBLE PLAYBOOK 管理主机组

要了解更多有关 身份管理(IdM)中的主机组 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

## 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. 使用 ANSIBLE PLAYBOOK 确保存在 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

步骤

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。

## 46.3. 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 确保主机组中的主机在身份管理(IdM)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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

- 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。

步骤

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。

### 46.4. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。
- 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。

步骤

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。

## 46.5. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- 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。

步骤

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 命令来验证 group_name 组是否包含 example_member，并且 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
```

**其他资源**

- 请参阅 ipa hostgroup-add-member-manager --help。
- 请参阅 ipa man page。

## 46.6. 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- 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。

### 步骤

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。

## 46.7. 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 确保外部主机组的嵌套主机组在身份管理(IdM)中的不存在。

### 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。
- 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。

步骤

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:
```

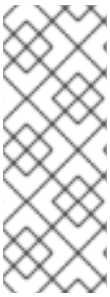
3. 显示应当缺少嵌套主机组的主机组的信息：

```
$ 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。

## 46.8. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

步骤

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。

## 46.9. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。

- 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。

步骤

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 命令来验证 group_name 组是否不包含 example_member 或不包含 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
```

**其他资源**

- 请参阅 ipa hostgroup-add-member-manager --help。
- 请参阅 ipa man page。

# 第 47 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哪些服务或服务组中的哪些服务访问指定的主机或主机组。例如，您可以使用 HBAC 规则实现以下目标：

- 将您域中对指定系统的访问权限限制到特定用户组的成员。
- 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，对于每个用户，该规则允许通过整个 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。

## 47.1. 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 规则之前不要禁用 allow_all 规则，因为如果这样做了，任何用户都将无法访问任何主机。

### 47.1.1. 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。

**步骤**

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 > 移到 Prospective 用户列表中，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 将 HBAC 规则应用到所有服务。

**注意**

默认情况下，只为 HBAC 规则配置最常用服务和服务组。

- 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**。

要添加更多的服务和服务组，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 规则 配置页面上所做的任何更改，请点击页面顶部的 **Save**。

### 47.1.2. 在 IdM WebUI 中测试 HBAC 规则

IdM 允许您使用模拟场景在各种情况下测试 HBAC 配置。执行这些模拟测试，您可以在在生产环境中部署 HBAC 规则之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只前，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，所以在模拟 HBAC 场景时，IdM 无法解析 AD 用户的组成员资格。

**流程**

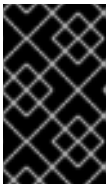
1. 选择 **Policy>Host-Based Access Control>HBAC Test**。
2. 在 **Who** 窗口上，指定您要在其身份下执行测试的用户，然后点 **Next**。

3. 在 Accessing 窗口上，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 窗口上，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 以对状态为 Enabled 的所有规则运行测试。选择 Include Disabled 以对状态为 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，则用户没有在测试中授予访问权限。
   - 如果您看到 ACCESS GRANTED，则用户可以成功访问主机。

   默认情况下，在显示测试结果时，IdM 会列出所有测试过的 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。

### 47.1.3. 在 IdM WebUI 中禁用 HBAC 规则

您可以禁用 HBAC 规则，但它只禁用该规则，且不删除它。如果禁用 HBAC 规则，您可以以后重新启

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。

## 47.2. 使用 CLI 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 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它，则拒绝所有用户对所有主机的访问。

### 47.2.1. 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。出于本例目的，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. 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，则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：

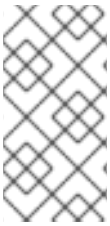
```
$ ipa hbacrule-mod rule_name --hostcat=all
-----------------------------
Modified HBAC rule "rule_name"
-----------------------------
  Rule name: rule_name
  Host category: all
  Enabled: TRUE
  Users: sysadmin
```

**注意**

如果 HBAC 规则与单个主机或组关联，则 ipa hbacrule-mod --hostcat=all 会失败。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，则 ipa hbacrule-mod --servicecat=all 会失败。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。

**其他资源**

- 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。

### 47.2.2. 在 IdM CLI 中测试 HBAC 规则

IdM 允许您使用模拟场景在各种情况下测试 HBAC 配置。执行这些模拟测试，您可以在在生产环境中部署 HBAC 规则之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只前，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，所以在模拟 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。您有选择测试单个 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. 添加第二个规则，仅允许 *sysadmin* 在所有主机上使用 ssh ：

```
$ 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 列出允许成功访问的规则，而 Not matched 列出阻止访问的规则。请注意，如果您没有指定 --rules 选项，则应用所有规则。使用 --rules 可独立测试每个规则。

**其他资源**

- 如需更多信息，请参阅 ipa hbactest --help。

### 47.2.3. 在 IdM CLI 中禁用 HBAC 规则

您可以禁用 HBAC 规则，但它只禁用该规则，且不删除它。如果禁用 HBAC 规则，您可以以后重新启用它。

**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"
-----------------------------
```

**其他资源**

- 如需了解更多详细信息，请参阅 ipa hbacrule-disable --help。

## 47.3. 为自定义 HBAC 服务添加 HBAC 服务条目

默认为 HBAC 规则配置最常用服务和服务组，但您也可以将任何其他可插入身份验证模块(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对域中的其他资源可用，但不允许在 HBAC 规则中使用服务。

### 47.3.1. 在 IdM Web UI 中为自定义 HBAC 服务添加 HBAC 服务条目

要添加自定义 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 HBAC 服务条目。
3. 输入服务的名称，然后单击 Add。

### 47.3.2. 在 IdM CLI 中为自定义 HBAC 服务添加 HBAC 服务条目

要添加自定义 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
```

**其他资源**

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

### 47.4. 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。

#### 47.4.1. 在 IdM WebUI 中添加 HBAC 服务组

要在 IdM WebUI 中添加 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. 在服务组配置页面上，点 Add 将 HBAC 服务添加为组成员。

#### 47.4.2. 在 IdM CLI 中添加 HBAC 服务组

要在 IdM CLI 中添加 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 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
------------------------
```

**其他资源**

- 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 48 章 使用 ANSIBLE PLAYBOOK 确保在 IDM 中存在基于主机的访问控制规则

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。

## 48.1. IDM 中的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到特定用户组的成员。
- 仅允许使用特定服务访问域中的系统。

默认情况下，使用一个名为 allow_all 的默认 HBAC 规则对IdM 进行配置，这意味着可通过整个 IdM 域中每个相关服务对每个用户的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。

## 48.2. 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，您已创建了一个带有 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 ipaadmin_password。

- 目标节点,也就是在其上执行 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。
- IdM 中您要用于 HBAC 规则的用户和用户组已存在。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。
- IdM 中要对其应用 HBAC 规则的主机和主机组已存在。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，其定义您要确保的 HBAC 策略存在。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. 以管理员的身份登录到 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。

**其他资源**

- 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 49 章 管理用户和主机的公共 SSH 密钥

SSH (Secure Shell)是一种协议，它在使用客户端-服务器架构的两个系统之间提供安全通信。SSH 允许用户远程登录到服务器主机系统，还允许一台主机访问另一台机器。

## 49.1. 关于 SSH 密钥格式

IdM 接受以下两种 SSH 密钥格式：

- OpenSSH 样式的密钥
- 原始 RFC 4253 样式的密钥

请注意，IdM 会在将它们保存到 IdM LDAP 服务器之前，自动将 RFC 4253 样式的密钥转换为 OpenSSH 样式的密钥。

IdM 服务器可以从上传的密钥 blob 中识别密钥类型，如 RSA 或 DSA 密钥。在密钥文件中，如 ~/.ssh/known_hosts，密钥条目通过服务器的主机名和 IP 地址、其类型和密钥来标识。例如：

```
host.example.com,1.2.3.4 ssh-rsa AAA...ZZZ==
```

这与用户公钥条目不同，后者具有 *type key== comment* 顺序的元素：

```
"ssh-rsa ABCD1234...== ipaclient.example.com"
```

密钥文件（如 id_rsa.pub ）包含三个部分：密钥类型、密钥以及额外的注释或标识符。将密钥上传到 IdM 时，您可以上传所有三个密钥部分或只上传密钥。如果您只上传密钥，IdM 会自动从上传的密钥中识别密钥类型，如 RSA 或 DSA。

如果使用 ~/.ssh/known_hosts 文件中的主机公钥条目，您必须重新排序它，以匹配用户密钥的格式， *type key== comment* ：

```
ssh-rsa AAA...ZZZ== host.example.com,1.2.3.4
```

IdM 可以从公钥的内容自动决定密钥类型。注释是可选的，以便更轻松地识别单个密钥。唯一必需的元

素是公钥 blob。

IdM 使用存储在以下 OpenSSH 样式文件中的公钥：

- 主机公钥位于 known_hosts 文件中。
- 用户公钥位于 authorized_keys 文件中。

其他资源

- 请参阅 RFC 4716
- 请参阅 RFC 4253

## 49.2. 关于 IDM 和 OPENSSH

在 IdM 服务器或客户端安装过程中，作为安装脚本的一部分：

- 在 IdM 客户端机器上配置了 OpenSSH 服务器和客户端。
- SSSD 被配置为在缓存中存储和检索用户和主机 SSH 密钥。这允许 IdM 充当 SSH 密钥的通用和集中的存储库。

如果您在客户端安装过程中启用了 SSH 服务，则在 SSH 服务第一次启动时会创建一个 RSA 密钥。

注意

当您运行 ipa-client-install 安装脚本将机器添加为 IdM 客户端时，会使用两个 SSH 密钥 RSA 和 DSA 创建客户端。

作为安装的一部分，您可以配置以下内容：

- 使用 --ssh-trust-dns 选项，将 OpenSSH 配置为自动信任存储密钥指纹的 IdM DNS 记录。
- 禁用 OpenSSH ，并防止安装脚本使用 --no-sshd 选项配置 OpenSSH 服务器。
- 防止主机使用 --no-dns-sshfp 选项创建带有其自身 DNS 条目的 DNS SSHFP 记录。

如果您在安装过程中没有配置服务器或客户端，您可以稍后手动配置 SSSD。有关如何手动配置 SSSD 的详情，请参考 配置 SSSD 以为 OpenSSH 服务提供缓存。请注意，SSSD 缓存的 SSH 密钥需要本地机器上的管理特权。

### 49.3. 生成 SSH 密钥

您可以使用 OpenSSH ssh-keygen 工具生成 SSH 密钥。

**流程**

1. 要生成 RSA SSH 密钥，请运行以下命令：

```
$ ssh-keygen -t rsa -C user@example.com
Generating public/private rsa key pair.
```

请注意，如果生成主机密钥，请将 user@example.com 替换为所需的主机名，如 server.example.com,1.2.3.4。

2. 指定要保存密钥的文件，或者按 enter 键接受显示的默认位置。

```
Enter file in which to save the key (/home/user/.ssh/id_rsa):
```

请注意，如果生成主机密钥，请将密钥保存到与用户的 ~/.ssh/ 目录不同的位置，这样您就不会覆盖任何现有的密钥。例如 /home/user/.ssh/host_keys。

3. 为您的私钥指定密码短语，或者按 enter 将密码短语留空。

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

```
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ONxjcMX7hJ5zly8F8ID9fpbqcuxQK+ylVLKDMsJPxGA user4@example.com
The key's randomart image is:
+---[RSA 3072]----+
|        ..o      |
|        .o +     |
|     E. .  o =   |
|    ..o=  o . +  |
|      +oS. = + o.|
|    . .o .* B =.+|
|     o + . X.+.= |
|      + o o.*+. .|
|       .   o=o . |
+----[SHA256]-----+
```

要上传此 SSH 密钥，请使用存储在显示的文件中的公钥字符串。

## 49.4. 管理主机的公用 SSH 密钥

OpenSSH 使用公钥来验证主机。一个机器尝试访问另一台计算机，并显示其密钥对。主机第一次验证时，目标机器上的管理员必须手动批准请求。然后，机器将主机的公钥存储在 known_hosts 文件中。每当远程机器再次尝试访问目标机器时，目标机器会检查其 known_hosts 文件，然后自动将访问权限授予批准的主机。

### 49.4.1. 使用 IdM Web UI 为主机上传 SSH 密钥

身份管理允许您将公共 SSH 密钥上传到主机条目。OpenSSH 使用公钥来验证主机。

**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. 您可以从 ~/.ssh/known_hosts 文件检索主机的密钥。例如：

```
server.example.com,1.2.3.4 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEApvjBvSFSkTU0WQW4eOweeo0DZZ08F9Ud21xI
Ly6FOhzwpXFGlyxvXZ52+siHBHbbqGL5+14N7UvElruyslIHx9LYUR/pPKSMXCGyboLy5
aTNl5OQ5EHwrhVnFDIKXkvp45945R7SKYCUtRumm0Iw6wq0XD4o+ILeVbV3wmcB1bX
```

```
s36ZvC/M6riefn9PcJmh6vNCvIsbMY6S+FhkWUTTiOXJjUDYRLIwM273FfWhzHK+SSQX
eBp/zIn1gFvJhSZMRi9HZpDoqxLbBB9QIdIw6U4MIjNmKsSl/ASpkFm2GuQ7ZK9KuMItY
2AoCuIRmRAdF8iYNHBTXNfFurGogXwRDjQ==
```

您也可以生成主机密钥。请参阅 生成 SSH 密钥。

2. 从密钥文件复制公钥。完整的密钥条目的格式是 host name,IP type key==。仅需要 key==，但您可以存储整个条目。要使用条目中的所有元素，请重新安排该条目，以便其具有顺序 type key== [host name,IP]。

```
cat /home/user/.ssh/host_keys.pub
ssh-rsa AAAAB3NzaC1yc2E...tJG1PK2Mq++wQ== server.example.com,1.2.3.4
```

3. 登录到 IdM Web UI。

4. 进入到 Identity>Hosts 选项卡。

5. 单击要编辑的主机名称。

6. 在 Host Settings 部分中，点 SSH 公钥 Add 按钮。

7. 将主机的公钥粘贴到 SSH public key 字段。

8. 点 Set。

9. 点 IdM Web UI 窗口顶部的 Save。

验证

- 在 Hosts Settings 部分下，验证密钥是否列在 SSH public keys 下。

### 49.4.2. 使用 IdM CLI 上传主机的 SSH 密钥

身份管理允许您将公共 SSH 密钥上传到主机条目。OpenSSH 使用公钥来验证主机。当使用 host-

add 创建主机或稍后修改条目时，主机 SSH 密钥会被添加到 IdM 中的主机条目中。

请注意，RSA 和 DSA 主机密钥是通过 ipa-client-install 命令创建的，除非 SSH 服务在安装脚本中被明确禁用。

**先决条件**

- 管理 IdM 或用户管理员角色的管理员特权。

**流程**

1. 使用 --sshpubkey 选项运行 host-mod 命令，来将 base64 编码的公钥上传到主机条目。

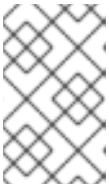
   因为添加主机密钥会更改主机的 DNS Secure Shell 指纹(SSHFP)记录，因此请使用 --updatedns 选项更新主机的 DNS 条目。例如：

   ```
   $ ipa host-mod --sshpubkey="ssh-rsa RjlzYQo==" --updatedns host1.example.com
   ```

   实际密钥通常以等号(=)结束，但更长。

2. 要上传多个密钥，请输入多个 --sshpubkey 命令行参数：

   ```
   --sshpubkey="RjlzYQo==" --sshpubkey="ZEt0TAo=="
   ```

   **注意**

   主机可以有多个公钥。

3. 上传主机密钥后，将 SSSD 配置为使用身份管理作为其身份域之一，并设置 OpenSSH，以使用 SSSD 工具管理主机密钥，如 配置 SSSD 以为 OpenSSH 服务提供缓存 中所述。

验证

- 运行 ipa host-show 命令来验证 SSH 公钥是否与指定的主机关联：

  ```
  $ ipa host-show client.ipa.test
  ```

```
...
SSH public key fingerprint:
SHA256:qGaqTZM60YPFTngFX0PtNPCKbluudwf1D2LqmDeOcuA
                        client@IPA.TEST (ssh-rsa)
...
```

### 49.4.3. 使用 IdM Web UI 删除主机的 SSH 密钥

您可以在主机密钥过期或者不再有效后删除它们。按照以下步骤，使用 IdM Web UI 删除单独的主机密钥。

**先决条件**

- 管理 IdM Web UI 或主机管理员角色的管理员特权。

**流程**

1. 登录到 IdM Web UI。
2. 进入到 Identity>Hosts 选项卡。
3. 单击要编辑的主机名称。
4. 在 Host Settings 部分下，点击您要删除的 SSH 公钥旁边的 Delete。
5. 点页面顶部的 Save。

验证

- 在 Host Settings 部分下，验证密钥是否不再列在 SSH public keys 下。

### 49.4.4. 使用 IdM CLI 删除主机的 SSH 密钥

您可以在主机密钥过期或者不再有效后删除它们。按照以下流程，使用 IdM CLI 删除单独的主机密钥。

**先决条件**

- 管理 IdM CLI 或主机管理员角色的管理员特权。

**流程**

- 要删除分配给主机帐户的所有 SSH 密钥，请将 --sshpubkey 选项添加到 ipa host-mod 命令中，而不指定任何密钥：

```
$ kinit admin
$ ipa host-mod --sshpubkey= --updatedns host1.example.com
```

请注意，最好使用 --updatedns 选项来更新主机的 DNS 条目。

如果上传的密钥中不包括类型，IdM 回自动从密钥中决定密钥类型。

验证

- 运行 ipa host-show 命令来验证 SSH 公钥是否不再与指定的主机关联：

```
ipa host-show client.ipa.test
  Host name: client.ipa.test
  Platform: x86_64
  Operating system: 4.18.0-240.el8.x86_64
  Principal name: host/client.ipa.test@IPA.TEST
  Principal alias: host/client.ipa.test@IPA.TEST
  Password: False
  Member of host-groups: ipaservers
  Roles: helpdesk
  Member of netgroups: test
  Member of Sudo rule: test2
  Member of HBAC rule: test
  Keytab: True
  Managed by: client.ipa.test, server.ipa.test
  Users allowed to retrieve keytab: user1, user2, user3
```

## 49.5. 管理用户的公共 SSH 密钥

身份管理允许您将公共 SSH 密钥上传到用户条目。有权访问相应私有 SSH 密钥的用户可以使用 SSH 登录到 IdM 机器，而无需使用 Kerberos 凭证。请注意，如果用户从没有 SSH 密钥文件的机器登录，则用户仍然可以通过提供 Kerberos 凭据进行身份验证。

### 49.5.1. 使用 IdM Web UI 为用户上传 SSH 密钥

身份管理允许您将公共 SSH 密钥上传到用户条目。有权访问相应私有 SSH 密钥的用户可以使用 SSH 登录到 IdM 机器，而无需使用 Kerberos 凭证。

**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. 进入到 Identity>Users 选项卡。
3. 单击要编辑的用户的名称。
4. 在 Account Settings 部分中，点 SSH public keys Add 按钮。
5. 将 Base 64 编码的公钥字符串粘贴到 SSH public key 字段中。
6. 点 Set。
7. 点 IdM Web UI 窗口顶部的 Save。

验证

- 在 Accounts Settings 部分下，验证密钥是否列在 SSH public keys 下。

### 49.5.2. 使用 IdM CLI 为用户上传 SSH 密钥

身份管理允许您将公共 SSH 密钥上传到用户条目。有权访问相应私有 SSH 密钥的用户可以使用 SSH 登录到 IdM 机器，而无需使用 Kerberos 凭证。

**先决条件**

- 管理 IdM CLI 或用户管理员角色的管理员特权。

**流程**

1. 运行带有 --sshpubkey 选项的 ipa user-mod 命令，将 base64 编码的公钥上传到用户条目。

```
$ ipa user-mod user --sshpubkey="ssh-rsa AAAAB3Nza...SNc5dv==
client.example.com"
```

请注意，在这个示例中，您将密钥类型、密钥和主机名标识符上传到用户条目。

2. 要上传多个密钥，请多次使用 --sshpubkey。例如，上传两个 SSH 密钥：

```
--sshpubkey="AAAAB3Nza...SNc5dv==" --sshpubkey="RjlzYQo...ZEt0TAo="
```

3. 要使用命令重定向并指向包含密钥的文件而不是手动粘贴密钥字符串，请使用以下命令：

```
ipa user-mod user --sshpubkey="$(cat ~/.ssh/id_rsa.pub)" --sshpubkey="$(cat
~/.ssh/id_rsa2.pub)"
```

验证

- 运行 ipa user-show 命令来验证 SSH 公钥是否与指定的用户关联：

```
$ ipa user-show user
User login: user
  First name: user
  Last name: user
  Home directory: /home/user
  Login shell: /bin/sh
  Principal name: user@IPA.TEST
  Principal alias: user@IPA.TEST
  Email address: user@ipa.test
  UID: 1118800019
  GID: 1118800019
  SSH public key fingerprint:
SHA256:qGaqTZM60YPFTngFX0PtNPCKbluudwf1D2LqmDeOcuA
                        user@IPA.TEST (ssh-rsa)
  Account disabled: False
  Password: False
```

```
Member of groups: ipausers
Subordinate ids: 3167b7cc-8497-4ff2-ab4b-6fcb3cb1b047
Kerberos keys available: False
```

### 49.5.3. 使用 IdM Web UI 删除用户的 SSH 密钥

按照以下流程，在 IdM Web UI 中从用户配置文件中删除 SSH 密钥。

**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. 进入到 Identity>Users 选项卡。
3. 单击要编辑的用户的名称。
4. 在 Account Settings 部分下，在 SSH public key 下，单击您要删除的密钥旁边的 Delete。
5. 点页面顶部的 Save。

**验证**

- 在 Account Settings 部分下，验证密钥是否不再列在 SSH public keys 下。

### 49.5.4. 使用 IdM CLI 删除用户的 SSH 密钥

按照以下流程，使用 IdM CLI 从用户配置文件中删除 SSH 密钥。

**先决条件**

- 管理 IdM CLI 或用户管理员角色的管理员特权。

**流程**

1. 要删除分配给用户帐户的所有 SSH 密钥，请在 ipa user-mod 命令中添加 --sshpubkey 选项，而不指定任何密钥：

```
$ ipa user-mod user --sshpubkey=
```

2. 要只删除特定的 SSH 密钥，请使用 --sshpubkey 选项指定您要保留的密钥，省略您要删除的密钥。

验证

- 运行 ipa user-show 命令来验证 SSH 公钥是否不再与指定的用户关联：

```
$ ipa user-show user
User login: user
  First name: user
  Last name: user
  Home directory: /home/user
  Login shell: /bin/sh
  Principal name: user@IPA.TEST
  Principal alias: user@IPA.TEST
  Email address: user@ipa.test
  UID: 1118800019
  GID: 1118800019
  Account disabled: False
  Password: False
  Member of groups: ipausers
  Subordinate ids: 3167b7cc-8497-4ff2-ab4b-6fcb3cb1b047
  Kerberos keys available: False
```

# 第 50 章 配置域解析顺序来解析简短的 AD 用户名

默认情况下，您必须以 user_name@domain.com 或 domain.com\user_name 格式指定完全限定域名，以便从 Active Directory(AD)环境中解析和验证用户和组。以下小节介绍了如何配置 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 服务器上设置全局域解析顺序
- 为 IdM 服务器上的 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 客户端上的 SSSD 中设置域解析顺序

## 50.1. 域解析顺序的工作方式

在带有 Active Directory (AD) 信任的身份管理 (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 用于来自 idm.example.com 域的 IdM 用户
- *<ad_username>*@ad.example.com 用于来自 ad.example.com 域的 AD 用户

默认情况下，如果您使用 *短名称* 格式执行用户或组查找，如 ad_username，IdM 仅搜索 IdM 域，且无法找到 AD 用户或组。要使用短名称解析 AD 用户或组，请通过设置 domain resolution order 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 配置中集中设置域解析顺序。IdM 按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，且您想要向不使用此选项指定的域发出请求，则必须使用完全限定的用户名。
- 如果您使用 domain resolution order 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 当域解析顺序设置了 AD 用户的 Pagure bug report Inconsistent compat user 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。对此选项使用非默认值会改变如何显示用户名，并可能会破坏 IdM 环境中的查找。

其他资源

- 传统 Linux 客户端的活动目录信任.

## 50.2. 在 IDM 服务器上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。这个示例设置域解析顺序按以下顺序搜索用户和组：

1. Active Directory (AD) 根域 ad.example.com
2. AD 子域 subdomain1.ad.example.com

3. IdM 域 idm.example.com

**先决条件**

- 已使用 AD 环境配置了信任。

**流程**

- 使用 ipa config-mod --domain-resolution-order 命令来按照您首选的顺序列出要搜索的域。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

- 验证您只能使用短名称从 ad.example.com 域检索用户信息。

```
[root@client ~]# id <ad_username>
uid=1916901102(ad_username) gid=1916900513(domain users)
groups=1916900513(domain users)
```

## 50.3. 为 IDM 服务器上的 ID 视图设置域解析顺序

此流程为 ID 视图设置域解析顺序，您可以将其应用到一组特定的 IdM 服务器和客户端。本例为 IdM 主机 client1.idm.example.com 创建一个名为 ADsubdomain1_first 的 ID 视图，并按照以下顺序设置域解析顺序搜索用户和组：

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

ID 视图中设置的域解析顺序会覆盖全局域解析顺序，但不覆盖 SSSD 配置中本地设置的任何域解析顺序。

先决条件

- 已使用 AD 环境配置了信任。

流程

1. 使用 --domain-resolution-order 选项创建 ID 视图。

```
[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
```

- 验证您只能使用短名称从 subdomain1.ad.example.com 域检索用户信息。

```
[root@client1 ~]# id <user_from_subdomain1>
uid=1916901106(user_from_subdomain1) gid=1916900513(domain users)
groups=1916900513(domain users)
```

## 50.4. 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*/ 目录中创建了一个带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 您使用 RHEL 9.4 或更高版本。
  - 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,也就是在其上执行 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。

**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、存储保护 secret.yml 文件的密码，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)
```

其他资源

- ansible-freeipa 上游文档中的 idview 模块

## 50.5. 在 IDM 客户端上的 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会覆盖任何全局和 ID 视图域解析顺序。

**先决条件**

- 已使用 AD 环境配置了信任。

**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**

- 验证您只能使用短名称从 subdomain1.ad.example.com 域检索用户信息。

```
[root@client2 ~]# id <user_from_subdomain1>
uid=1916901106(user_from_subdomain1) gid=1916900513(domain users)
groups=1916900513(domain users)
```

## 50.6. 其他资源

- 使用 ID 视图来覆盖 IdM 客户端上的用户属性值

# 第 51 章 在 IDM 中使用 AD USER PRINCIPAL NAMES 启用身份验证

## 51.1. IDM 信任的 AD 林中的用户主体名称

作为 Identity Management(IdM)管理员，您可以允许 AD 用户使用替代的 User Principal Names (UPN)来访问 IdM 域中的资源。UPN 是 AD 用户以 user_name@KERBEROS 格式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 设置替代值，因为您可以在 AD 林中配置额外的 Kerberos 别名和 UPN 后缀。

例如，如果某个公司使用 Kerberos 域 AD.EXAMPLE.COM，则用户的默认 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 UPN。如果您的公司最近遇到了合并并且您希望为用户提供统一登录命名空间，备用 *UPN （也称为企业 UPN*）特别方便。

UPN 后缀仅在 AD 林根中定义时对 IdM 可见。作为 AD 管理员，您可以使用 Active Directory Domain and Trust 工具程序或 PowerShell 命令行工具定义 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 Domain 和 Trust 实用程序。

红帽建议不要通过低级别修改来配置 UPN，如使用 ldapmodify 命令为用户设置 用户 PrincipalName 属性，因为 Active Directory 不会验证这些操作。

在 AD 端定义了新的 UPN 后，在 IdM 服务器上运行 ipa trust-fetch-domains 命令以检索更新的 UPN。请参阅确保 AD UPN 是 IdM 中的最新状态。

IdM 将一个域的 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。

其他资源

- 如何在 AD 林根中脚本 UPN 后缀设置

- 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

## 51.2. 确保 AD UPN 在 IDM 中是最新的

在受信任的 Active Directory(AD)林中添加或删除 User Principal Name(UPN)后缀后，刷新 IdM 服务器上受信任林的信息。

**先决条件**

- IdM 管理员凭证。

**流程**

- 输入 ipa trust-fetch-domains 命令。请注意，预期会出现空输出：

```
[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 命令来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 realm 条目的一部分。

## 51.3. 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，您可以使用这些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 权限来修改以下配置文件，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。如果您没有该文件，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 AD 域检索信任信息：

```
[root@server ~]# ipa trust-fetch-domains <ad.example.com>
```

9. 在以下日志文件中查看调试输出和故障排除信息：

- /var/log/httpd/error_log
- /var/log/samba/log.*

**其他资源**

- 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 52 章 启用 AD 用户管理 IDM

## 52.1. AD 用户的 ID 覆盖

您可以集中管理 POSIX 环境中的 Active Directory(AD)用户和组到 POSIX 环境中的 Identity Management(IdM)资源访问，方法是作为 IdM 组的成员为 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。使用这个功能，IdM LDAP 服务器可以将 IdM 组的访问控制规则应用到 AD 用户。

AD 用户可以使用 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。

**其他资源**

- 为活动目录用户使用 ID 视图

## 52.2. 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，在配置为信任控制器或信任代理的 IdM 服务器上工作。

**先决条件**

- 设定了一个正常工作的 IdM 环境。详情请参阅 安装身份管理。

- 设置 IdM 环境和 AD 之间的工作信任。

**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 创建一个 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，因为它与活动目录进行交互。

   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 时会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，您可以向角色添加 ID 覆盖，如 User Administrator 角色：

```
# ipa role-add-member 'User Administrator' --
idoverrideusers=ad_user@ad.example.com
```

**其他资源**

- 为活动目录用户使用 ID 视图

## 52.3. 使用 ANSIBLE 启用 AD 用户管理 IDM

按照以下流程，使用 Ansible playbook 确保用户 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。
- 您要添加用户 ID 覆盖的组已在 IdM 中存在。
- 您可以使用 IdM 或更高版本的 4.8.7 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，您已创建了一个带有 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 ipaadmin_password。
- 目标节点,也就是在其上执行 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。

**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 admin 密码。
- admins 是您要添加 ad_user@ad.example.com ID 覆盖的 IdM POSIX 组的名称。此组中的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、存储保护 secret.yml 文件的密码，以及清单文件：

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

其他资源

- 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

## 52.4. 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)，并运行适用于其角色的命令。

1. 销毁 IdM 管理员的当前 Kerberos 票据：

```
# kdestroy -A
```

**注意**

需要 Kerberos 票据的破坏性，因为 MIT Kerberos 中的 GSSAPI 实现根据首选从目标服务的域中选择凭证，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 覆盖是否从与该组中的任何 IdM 用户身份获取相同的特权。如果 AD 用户的 ID 覆盖已添加到 admins 组中，则 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
```

## 52.5. 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 9.4 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个带有 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 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,也就是在其上执行 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。

**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、存储保护 secret.yml 文件的密码，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
[...]
```

**其他资源**

- idoverrideuser 和 ipagroup ansible-freeipa 上游文档
- 启用 AD 用户管理 IdM

# 第 53 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 9.1 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得到带有 Kerberos 票据的 RHEL 身份管理(IdM)单点登录能力。

主要特性包括：

- 使用 ipa idp-* 命令添加、修改和删除到外部 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。

## 53.1. 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。

## 53.2. 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 客户端会输出命令行中的链接和代码。

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 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。
- 通过 logind 服务，使用控制台本地登录。
- 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。

**其他资源**

- 对外部身份提供程序进行身份验证

- RFC 8628：OAuth 2.0 设备授权

### 53.3. 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 作为 OAuth 应用程序注册到外部 IdP，并获取了客户端 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 9.1 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
```

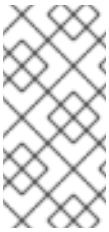
**其他资源**

- 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

### 53.4. 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>* |

**其他资源**

- 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

### 53.5. 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**

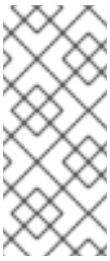
Keycloak 和 Okta 所需的 IdP 模板的基本 URL

**--organization**

Microsoft Azure 所需的 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 客户端中的 secret，则使用这个选项。如果您在创建 IdP 引用时使用这个选项，则会以交互方式会提示您输入 secret。将客户端 secret 作为密码保护。



**注意**

RHEL 9.1 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 9.2 及之后的版本中使用 SSSD。

**其他资源**

- 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

## 53.6. 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 9.1 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
```

## 53.7. 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 9.1 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
```

### 53.8. 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 9.1 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，并确认对于带有外部 IdP 的预身份验证的行 config: pa_type 显示 152。

```
[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。

## 53.9. 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 9.1 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，并确认对于带有外部 IdP 的预身份验证的行 config: pa_type 显示 152。

```
[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
```

## 53.10. 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 Red Hat Single Sign-On (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 email |
| --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 email** |
| **--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 **${ipaidpbaseurl}**，而您为 **--organization** 指定的选项替换字符串 **`${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 email** |
| **--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 email** |

| 选项 | 值 |
| --- | --- |
| --idp-user-id=STR | email |

**其他资源**

- 预填充的 IdP 模板

# 第 54 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 RHEL 9.1 或更高版本中提供的 SSSD 版本进行身份验证，在外部 IdP 执行身份验证和授权后，它们会收到带有 Kerberos 票据的 RHEL Identity Management (IdM)单点登录功能。

## 54.1. 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。

## 54.2. 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 客户端会输出命令行中的链接和代码。
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 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。
- 通过 logind 服务，使用控制台本地登录。
- 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。

**其他资源**

- 对外部身份提供程序进行身份验证
- RFC 8628：OAuth 2.0 设备授权

### 54.3. 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 9.1 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 9.4 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个带有 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 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、存储保护 secret.yml 文件的密码，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

其他资源

- idp ansible-freeipa 上游文档

## 54.4. 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 9.1 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 9.4 或更高版本。
  - 示例假定在 ~/*MyPlaybooks*/ 目录中，您已创建了一个带有 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 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、存储保护 secret.yml 文件的密码，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
```

**其他资源**

- idp ansible-freeipa 上游文档

## 54.5. 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 9.1 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，并确认对于带有外部 IdP 的预身份验证的行 config: pa_type 显示 152。

```
[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。

## 54.6. 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 9.1 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，并确认对于带有外部 IdP 的预身份验证的行 config: pa_type 显示 152。

```
[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
```

## 54.7. 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 Red Hat Single Sign-On (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 email** |
| **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 email** |
| **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 email |
| 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 email** |
| **idp_user_id: STR** | **email** |

**其他资源**

- **预填充的 IdP 模板**

# 第 55 章 在 IDM 中使用基于资源的受限委托

您可以使用基于资源的受限委托(RBCD)来委派对服务的访问。RBCD 允许资源级别上委派的细粒度控制，并且可由委派凭据的服务的所有者设置访问权限。例如，这在身份管理(IdM)和活动目录(AD)之间的集成中很有用。

自 2019 起，当目标和代理服务都属于不同的林时，Microsoft AD 强制使用 RBCD。

## 55.1. 其他资源

- 在 IdM 中使用受限委托

## 55.2. 在身份管理中基于资源的受限委托

基于资源的受限委托(RBCD)与常规受限委托在多个方面有所不同：

- 粒度：在 RBCD 中，委派在资源级别指定。
- 访问授权责任：在 RBCD 中，访问是由服务所有者，而不是 Kerberos 管理员控制的。

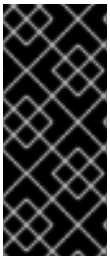
在常规受限委托中，用户到代理的服务(S4U2proxy)扩展代表用户获取其他服务的服务票据。第二个服务通常是在用户的授权上下文下，代表第一个服务执行任务的代理。使用受限委托无需用户委派其完整票据授予票 (TGT)。

身份管理(IdM)通常使用 Kerberos S4U2proxy 功能来允许 Web 服务器框架代表用户获取 LDAP 服务票据。

当 IdM 与活动目录(AD)集成时，IdM 框架也使用受限委托来代表用户对各种服务（包括 IdM 和活动目录端的 SMB 和 DCE RPC 端点）进行操作。

当 IdM 域中的应用程序需要代表用户对不同服务进行操作时，需要委派权限。在常规受限委托中，这需要域管理员明确创建一个规则，以允许第一个服务将用户凭据委派给下一个服务。使用 RBCD 时，可由委派凭据的服务的所有者创建委派权限。

对于 IdM-AD 集成，当两个服务都属于同一 IdM 域的一部分时，可以在 IdM 端授予 RBCD 权限。



**重要**

目前，只有 IdM 域中的服务才能使用 RBCD 规则进行配置。如果目标服务是 AD 域的一部分，则只能在 AD 端授予权限。因为 AD 域控制器无法解析 IdM 服务信息来创建规则，这目前还不支持。

## 55.3. 使用 RBCD 委派对服务的访问

要使用 RBCD 委派对服务的访问，必须在运行服务的主机上添加一个规则。这个示例流程描述了如何将用户凭据委派给带有 Kerberos 服务 HTTP/client.example.test 的 web 应用程序的文件服务器 nfs/client.example.test。您可以在 client.example.test 主机上执行此操作，因为主机始终管理在其上运行的服务。

**先决条件**

- 您可以访问 client.example.test 主机的 /etc/krb5.keytab 文件。
- nfs/client.example.test 服务 keytab 存在。
- HTTP/client.example.test 的 keytab /path/to/web-service.keytab 存在。

**流程**

1. 在 client.example.test 主机上，获取一个 Kerberos 票据：

```
# kinit -k
```

2. 定义 RBCD ACL：

```
# ipa service-add-delegation nfs/client.example.test HTTP/client.example.test

-----------------------------------------------------
Added new resource delegation to the service principal
"nfs/client.example.test@EXAMPLE.TEST"
```

```
-------------------------------------------------------
 Principal name: nfs/client.example.test@EXAMPLE.TEST
 Delegation principal: HTTP/client.example.test@EXAMPLE.TEST
```

验证

要验证委派是否被正确设置，您可以通过 HTTP 服务模拟 testuser 用户登录，并执行一个到 NFS 服务的协议转换。

1. 查看 NFS 服务，以验证委派规则是否存在：

```
# ipa service-show nfs/client.example.test

 Principal name: nfs/client.example.test@EXAMPLE.TEST
 Principal alias: nfs/client.example.test@EXAMPLE.TEST
 Delegation principal: HTTP/client.example.test@EXAMPLE.TEST
 Keytab: True
 Managed by: client.example.test
```

2. 为 HTTP 服务主体获取一个 Kerberos 票据：

```
# kinit -kt http.keytab HTTP/client.example.test
```

3. 验证票据授予票据是否存在：

```
# klist -f
Ticket cache: KCM:0:99799
Default principal: HTTP/client.example.test@EXAMPLE.TEST

Valid starting       Expires              Service principal
10/13/2023 14:39:23  10/14/2023 14:05:07  krbtgt/EXAMPLE.TEST@EXAMPLE.TEST
 Flags: FIA
```

4. 代表 testuser 执行协议转换：

```
# kvno -U testuser -P nfs/client.example.test
nfs/client.example.test@EXAMPLE.TEST: kvno = 1
```

5. 验证在协议转换过程中代表 testuser 获得的票据是否存在：

```
# klist -f
Ticket cache: KCM:0:99799
```

```
Default principal: HTTP/client.example.test@EXAMPLE.TEST

Valid starting       Expires              Service principal
10/13/2023 14:39:38  10/14/2023 14:05:07  HTTP/client.example.test@EXAMPLE.TEST
 for client testuser@EXAMPLE.TEST, Flags: FAT
10/13/2023 14:39:23  10/14/2023 14:05:07  krbtgt/EXAMPLE.TEST@EXAMPLE.TEST
 Flags: FIA
10/13/2023 14:39:38  10/14/2023 14:05:07  nfs/client.example.test@EXAMPLE.TEST
 for client testuser@EXAMPLE.TEST, Flags: FAT
```