



# Red Hat Enterprise Linux 9

## IdM 사용자가 호스트 및 액세스 제어 규칙 관리

사용자와 호스트 구성, 그룹에서 관리하고, 호스트 기반(HBAC) 및 RBAC(역할 기반 액세스 제어) 규칙으로 액세스 제어



## Red Hat Enterprise Linux 9 IdM 사용자가 호스트 및 액세스 제어 규칙 관리

---

사용자와 호스트 구성, 그룹에서 관리하고, 호스트 기반(HBAC) 및 RBAC(역할 기반 액세스 제어) 규칙으로 액세스 제어

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 법적 공지

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Managing\_IdM\_users\_groups\_hosts\_and\_access\_control\_rules.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 초록

이 문서 컬렉션은 Red Hat Enterprise Linux 8의 ID 관리에서 사용자, 그룹 및 호스트를 생성하고 HBAC 및 RBAC 규칙을 통해 해당 호스트에 대한 액세스를 관리하는 방법에 대한 지침을 제공합니다.

## 차례

|                                                              |           |
|--------------------------------------------------------------|-----------|
| 보다 포괄적 수용을 위한 오픈 소스 용어 교체 .....                              | 9         |
| RED HAT 문서에 관한 피드백 제공 .....                                  | 10        |
| <b>1장. IDM 명령줄 유틸리티 소개 .....</b>                             | <b>11</b> |
| 1.1. IPA 명령줄 인터페이스란 무엇입니까?                                   | 11        |
| 1.2. IPA 도움말이란 무엇입니까?                                        | 11        |
| 1.3. IPA 도움말 주제 사용                                           | 12        |
| 1.4. IPA 도움말 명령 사용                                           | 13        |
| 1.5. IPA 명령 구조                                               | 13        |
| 1.6. IPA 명령을 사용하여 IDM에 사용자 계정 추가                             | 15        |
| 1.7. IPA 명령을 사용하여 IDM에서 사용자 계정 수정                            | 17        |
| 1.8. IDM 유틸리티에 값 목록을 제공하는 방법                                 | 18        |
| 1.9. IDM 유틸리티와 특수 문자를 사용하는 방법                                | 19        |
| <b>2장. 명령줄을 사용하여 사용자 계정 관리 .....</b>                         | <b>21</b> |
| 2.1. 사용자 라이프 사이클                                             | 21        |
| 2.2. 명령줄을 사용하여 사용자 추가                                        | 23        |
| 2.3. 명령줄을 사용하여 사용자 활성화                                       | 25        |
| 2.4. 명령줄을 사용하여 사용자 보존                                        | 25        |
| 2.5. 명령줄을 사용하여 사용자 삭제                                        | 26        |
| 2.6. 명령줄을 사용하여 사용자 복원                                        | 27        |
| <b>3장. IDM 웹 UI를 사용하여 사용자 계정 관리 .....</b>                    | <b>29</b> |
| 3.1. 사용자 라이프 사이클                                             | 29        |
| 3.2. 웹 UI에서 사용자 추가                                           | 31        |
| 3.3. IDM 웹 UI에서 스테이징 사용자 활성화                                 | 34        |
| 3.4. 웹 UI에서 사용자 계정 비활성화                                      | 35        |
| 3.5. 웹 UI에서 사용자 계정 활성화                                       | 37        |
| 3.6. IDM 웹 UI에서 활성 사용자 유지                                    | 38        |
| 3.7. IDM 웹 UI에서 사용자 복원                                       | 39        |
| 3.8. IDM 웹 UI에서 사용자 삭제                                       | 40        |
| <b>4장. ANSIBLE 플레이북을 사용하여 사용자 계정 관리 .....</b>                | <b>43</b> |
| 4.1. 사용자 라이프 사이클                                             | 43        |
| 4.2. ANSIBLE 플레이북을 사용하여 IDM 사용자가 있는지 확인                      | 45        |
| 4.3. ANSIBLE 플레이북을 사용하여 여러 IDM 사용자가 있는지 확인                   | 47        |
| 4.4. ANSIBLE 플레이북을 사용하여 JSON 파일에서 여러 IDM 사용자가 있는지 확인         | 49        |
| 4.5. ANSIBLE 플레이북을 사용하여 사용자가 없는지 확인                          | 51        |
| 4.6. 추가 리소스                                                  | 53        |
| <b>5장. IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여 .....</b>         | <b>54</b> |
| 5.1. IDM 클라이언트의 SUDO 액세스                                     | 54        |
| 5.2. CLI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여           | 54        |
| 5.3. IDM 웹 UI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여      | 57        |
| 5.4. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 CLI에 SUDO 규칙 생성           | 61        |
| 5.5. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 IDM WEBUI에 SUDO 규칙 생성     | 64        |
| 5.6. IDM 클라이언트에서 SUDO에 대한 GSSAPI 인증 활성화                      | 71        |
| 5.7. IDM 클라이언트에서 SUDO에 대한 GSSAPI 인증 활성화 및 적용                 | 74        |
| 5.8. PAM 서비스의 GSSAPI 인증을 제어하는 SSSD 옵션                        | 77        |
| 5.9. SUDO용 GSSAPI 인증 문제 해결                                   | 79        |
| 5.10. ANSIBLE 플레이북을 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 확인 | 81        |

|                                                        |            |
|--------------------------------------------------------|------------|
| <b>6장. LDAPMODIFY를 사용하여 외부 IDM 사용자 관리</b>              | <b>85</b>  |
| 6.1. IDM 사용자 계정을 외부에서 관리하기 위한 템플릿                      | 85         |
| 6.2. IDM 그룹 계정을 외부에서 관리하기 위한 템플릿                       | 88         |
| 6.3. LDAPMODIFY를 사용하여 IDM 사용자 보존                       | 90         |
| <b>7장. 사용자의 외부 프로비저닝을 위해 IDM 구성</b>                    | <b>93</b>  |
| 7.1. 스테이지 사용자 계정의 자동 활성화를 위해 IDM 계정 준비                 | 93         |
| 7.2. IDM 단계 사용자 계정의 자동 활성화 구성                          | 96         |
| 7.3. LDIF 파일에 정의된 IDM 단계 사용자 추가                        | 99         |
| 7.4. LDAPMODIFY를 사용하여 CLI에서 직접 IDM 스테이지 사용자 추가         | 100        |
| 7.5. 추가 리소스                                            | 103        |
| <b>8장. CLI를 사용하여 IDM에서 셀프 서비스 규칙 관리</b>                | <b>104</b> |
| 8.1. IDM의 셀프 서비스 액세스 제어                                | 104        |
| 8.2. CLI를 사용하여 셀프 서비스 규칙 생성                            | 104        |
| 8.3. CLI를 사용하여 셀프 서비스 규칙 편집                            | 105        |
| 8.4. CLI를 사용하여 셀프 서비스 규칙 삭제                            | 106        |
| <b>9장. IDM 웹 UI를 사용하여 셀프 서비스 규칙 관리</b>                 | <b>108</b> |
| 9.1. IDM의 셀프 서비스 액세스 제어                                | 108        |
| 9.2. IDM 웹 UI를 사용하여 셀프 서비스 규칙 생성                       | 108        |
| 9.3. IDM 웹 UI를 사용하여 셀프 서비스 규칙 편집                       | 111        |
| 9.4. IDM 웹 UI를 사용하여 셀프 서비스 규칙 삭제                       | 112        |
| <b>10장. ANSIBLE 플레이북을 사용하여 IDM에서 셀프 서비스 규칙 관리</b>      | <b>114</b> |
| 10.1. IDM의 셀프 서비스 액세스 제어                               | 114        |
| 10.2. ANSIBLE을 사용하여 셀프 서비스 규칙이 있는지 확인합니다.              | 115        |
| 10.3. ANSIBLE을 사용하여 셀프 서비스 규칙이 없는지 확인                  | 117        |
| 10.4. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 속성이 있는지 확인합니다.       | 119        |
| 10.5. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 속성이 없는지 확인합니다.       | 121        |
| <b>11장. IDM CLI에서 사용자 그룹 관리</b>                        | <b>124</b> |
| 11.1. IDM의 다양한 그룹 유형                                   | 124        |
| 11.2. 직접 및 간접 그룹 멤버                                    | 126        |
| 11.3. IDM CLI를 사용하여 사용자 그룹 추가                          | 126        |
| 11.4. IDM CLI를 사용하여 사용자 그룹 검색                          | 127        |
| 11.5. IDM CLI를 사용하여 사용자 그룹 삭제                          | 128        |
| 11.6. IDM CLI를 사용하여 사용자 그룹에 멤버 추가                      | 128        |
| 11.7. 사용자 개인 그룹 없이 사용자 추가                              | 130        |
| 11.7.1. 사용자 개인 그룹이 없는 사용자                              | 130        |
| 11.7.2. 개인 그룹이 전역적으로 활성화된 경우 사용자 개인 그룹이 없는 사용자 추가      | 131        |
| 11.7.3. 모든 사용자에게 대해 사용자 개인 그룹 비활성화                     | 131        |
| 11.7.4. 사용자 개인 그룹이 전역적으로 비활성화될 때 사용자 추가                | 132        |
| 11.8. IDM CLI를 사용하여 IDM 사용자 그룹에 사용자 또는 그룹 추가           | 133        |
| 11.9. IDM CLI를 사용하여 그룹 멤버 보기                           | 134        |
| 11.10. IDM CLI를 사용하여 사용자 그룹에서 멤버 제거                    | 135        |
| 11.11. IDM CLI를 사용하여 IDM 사용자 그룹에서 멤버 관리자로 사용자 또는 그룹 제거 | 136        |
| <b>12장. IDM 웹 UI에서 사용자 그룹 관리</b>                       | <b>138</b> |
| 12.1. IDM의 다양한 그룹 유형                                   | 138        |
| 12.2. 직접 및 간접 그룹 멤버                                    | 140        |
| 12.3. IDM 웹 UI를 사용하여 사용자 그룹 추가                         | 140        |
| 12.4. IDM 웹 UI를 사용하여 사용자 그룹 삭제                         | 141        |
| 12.5. IDM 웹 UI를 사용하여 사용자 그룹에 멤버 추가                     | 142        |
| 12.6. 웹 UI를 사용하여 IDM 사용자 그룹에 사용자 또는 그룹 추가              | 143        |

|                                                              |            |
|--------------------------------------------------------------|------------|
| 12.7. IDM 웹 UI를 사용하여 그룹 멤버 보기                                | 146        |
| 12.8. IDM 웹 UI를 사용하여 사용자 그룹에서 멤버 제거                          | 147        |
| 12.9. 웹 UI를 사용하여 IDM 사용자 그룹에서 멤버 관리자로 사용자 또는 그룹 제거           | 148        |
| <b>13장. ANSIBLE 플레이북을 사용하여 사용자 그룹 관리</b>                     | <b>150</b> |
| 13.1. IDM의 다양한 그룹 유형                                         | 150        |
| 13.2. 직접 및 간접 그룹 멤버                                          | 152        |
| 13.3. ANSIBLE 플레이북을 사용하여 IDM 그룹 및 그룹 멤버가 있는지 확인              | 153        |
| 13.4. ANSIBLE 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 있는지 확인          | 155        |
| 13.5. ANSIBLE 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 없는지 확인          | 157        |
| <b>14장. IDM CLI를 사용하여 그룹 멤버십 자동화</b>                         | <b>159</b> |
| 14.1. 자동 그룹 멤버십의 이점                                          | 160        |
| 14.2. 자동 멤버 규칙                                               | 160        |
| 14.3. IDM CLI를 사용하여 AUTOMEMBER 규칙 추가                         | 161        |
| 14.4. IDM CLI를 사용하여 AUTOMEMBER 규칙에 조건 추가                     | 162        |
| 14.5. IDM CLI를 사용하여 기존 AUTOMEMBER 규칙 보기                      | 164        |
| 14.6. IDM CLI를 사용하여 AUTOMEMBER 규칙 삭제                         | 165        |
| 14.7. IDM CLI를 사용하여 AUTOMEMBER 규칙에서 조건 제거                    | 166        |
| 14.8. IDM CLI를 사용하여 기존 항목에 자동 멤버십 규칙 적용                      | 167        |
| 14.9. IDM CLI를 사용하여 기본 AUTOMEMBER 그룹 구성                      | 168        |
| <b>15장. IDM 웹 UI를 사용하여 그룹 멤버십 자동화</b>                        | <b>170</b> |
| 15.1. 자동 그룹 멤버십의 이점                                          | 171        |
| 15.2. 자동 멤버 규칙                                               | 171        |
| 15.3. IDM 웹 UI를 사용하여 AUTOMEMBER 규칙 추가                        | 172        |
| 15.4. IDM 웹 UI를 사용하여 자동 멤버 규칙에 조건 추가                         | 173        |
| 15.5. IDM 웹 UI를 사용하여 기존 AUTOMEMBER 규칙 및 조건 보기                | 175        |
| 15.6. IDM 웹 UI를 사용하여 AUTOMEMBER 규칙 삭제                        | 176        |
| 15.7. IDM 웹 UI를 사용하여 자동 멤버 규칙에서 조건 제거                        | 177        |
| 15.8. IDM 웹 UI를 사용하여 기존 항목에 자동 멤버십 규칙 적용                     | 178        |
| 15.8.1. 모든 사용자 또는 호스트에 대해 자동 멤버십 다시 작성                       | 179        |
| 15.8.2. 단일 사용자 또는 호스트에 대해서만 자동 멤버십 다시 작성                     | 179        |
| 15.9. IDM 웹 UI를 사용하여 기본 사용자 그룹 구성                            | 180        |
| 15.10. IDM 웹 UI를 사용하여 기본 호스트 그룹 구성                           | 181        |
| <b>16장. ANSIBLE을 사용하여 IDM의 그룹 멤버십 자동화</b>                    | <b>183</b> |
| 16.1. IDM 관리를 위한 ANSIBLE 제어 노드 준비                            | 183        |
| 16.2. ANSIBLE을 사용하여 IDM 사용자 그룹에 대한 AUTOMEMBER 규칙이 있는지 확인합니다. | 186        |
| 16.3. ANSIBLE을 사용하여 IDM 사용자 그룹 AUTOMEMBER 규칙에 지정된 조건이 있는지 확인 | 188        |
| 16.4. ANSIBLE을 사용하여 IDM 사용자 그룹 AUTOMEMBER 규칙에 조건이 없는지 확인합니다. | 191        |
| 16.5. ANSIBLE을 사용하여 IDM 사용자 그룹의 AUTOMEMBER 규칙이 없는지 확인합니다.    | 194        |
| 16.6. ANSIBLE을 사용하여 IDM 호스트 그룹 AUTOMEMBER 규칙에 조건이 있는지 확인합니다. | 197        |
| 16.7. 추가 리소스                                                 | 199        |
| <b>17장. IDM CLI를 사용하여 사용자를 관리할 수 있도록 사용자 그룹에 권한 위임</b>       | <b>201</b> |
| 17.1. 위임 규칙                                                  | 201        |
| 17.2. IDM CLI를 사용하여 위임 규칙 생성                                 | 201        |
| 17.3. IDM CLI를 사용하여 기존 위임 규칙 보기                              | 202        |
| 17.4. IDM CLI를 사용하여 위임 규칙 수정                                 | 203        |
| 17.5. IDM CLI를 사용하여 위임 규칙 삭제                                 | 204        |
| <b>18장. IDM WEBUI를 사용하여 사용자를 관리할 수 있도록 사용자 그룹에 권한 위임</b>     | <b>205</b> |
| 18.1. 위임 규칙                                                  | 205        |
| 18.2. IDM WEBUI를 사용하여 위임 규칙 생성                               | 205        |

|                                                           |            |
|-----------------------------------------------------------|------------|
| 18.3. IDM WEBUI를 사용하여 기존 위임 규칙 보기                         | 207        |
| 18.4. IDM WEBUI를 사용하여 위임 규칙 수정                            | 208        |
| 18.5. IDM WEBUI를 사용하여 위임 규칙 삭제                            | 210        |
| <b>19장. ANSIBLE 플레이북을 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한 위임</b> | <b>211</b> |
| 19.1. 위임 규칙                                               | 211        |
| 19.2. IDM용 ANSIBLE 인벤토리 파일 생성                             | 211        |
| 19.3. ANSIBLE을 사용하여 위임 규칙이 있는지 확인합니다.                     | 213        |
| 19.4. ANSIBLE을 사용하여 위임 규칙이 없는지 확인합니다.                     | 215        |
| 19.5. ANSIBLE을 사용하여 위임 규칙에 특정 속성이 있는지 확인                  | 217        |
| 19.6. ANSIBLE을 사용하여 위임 규칙에 특정 속성이 없는지 확인                  | 220        |
| <b>20장. CLI를 사용하여 IDM에서 역할 기반 액세스 제어 관리</b>               | <b>223</b> |
| 20.1. IDM의 역할 기반 액세스 제어                                   | 223        |
| 20.1.1. IdM의 권한                                           | 223        |
| 20.1.2. 기본 관리 권한                                          | 225        |
| 20.1.3. IdM의 권한                                           | 227        |
| 20.1.4. IdM의 역할                                           | 228        |
| 20.1.5. ID 관리에서 사전 정의된 역할                                 | 228        |
| 20.2. CLI에서 IDM 권한 관리                                     | 229        |
| 20.3. 기존 권한에 대한 명령 옵션                                     | 232        |
| 20.4. CLI에서 IDM 권한 관리                                     | 233        |
| 20.5. 기존 권한에 대한 명령 옵션                                     | 233        |
| 20.6. CLI에서 IDM 역할 관리                                     | 234        |
| 20.7. 기존 역할에 대한 명령 옵션                                     | 235        |
| <b>21장. IDM 웹 UI를 사용하여 역할 기반 액세스 제어 관리</b>                | <b>237</b> |
| 21.1. IDM의 역할 기반 액세스 제어                                   | 237        |
| 21.1.1. IdM의 권한                                           | 237        |
| 21.1.2. 기본 관리 권한                                          | 239        |
| 21.1.3. IdM의 권한                                           | 241        |
| 21.1.4. IdM의 역할                                           | 242        |
| 21.1.5. ID 관리에서 사전 정의된 역할                                 | 242        |
| 21.2. IDM 웹 UI에서 권한 관리                                    | 243        |
| 21.3. IDM WEBUI에서 권한 관리                                   | 249        |
| 21.4. IDM 웹 UI에서 역할 관리                                    | 253        |
| <b>22장. ANSIBLE 플레이북을 사용하여 IDM을 관리하기 위한 환경 준비</b>         | <b>259</b> |
| <b>23장. ANSIBLE 플레이북을 사용하여 IDM에서 역할 기반 액세스 제어 관리</b>      | <b>262</b> |
| 23.1. IDM의 권한                                             | 263        |
| 23.2. 기본 관리 권한                                            | 264        |
| 23.3. IDM의 권한                                             | 267        |
| 23.4. IDM의 역할                                             | 267        |
| 23.5. ID 관리에서 사전 정의된 역할                                   | 268        |
| 23.6. ANSIBLE을 사용하여 권한이 있는 IDM RBAC 역할이 있는지 확인            | 268        |
| 23.7. ANSIBLE을 사용하여 IDM RBAC 역할이 없는지 확인                   | 271        |
| 23.8. ANSIBLE을 사용하여 IDM RBAC 역할에 사용자 그룹이 할당되었는지 확인        | 273        |
| 23.9. ANSIBLE을 사용하여 특정 사용자가 IDM RBAC 역할에 할당되지 않았는지 확인     | 275        |
| 23.10. ANSIBLE을 사용하여 서비스가 IDM RBAC 역할의 멤버인지 확인            | 278        |
| 23.11. ANSIBLE을 사용하여 호스트가 IDM RBAC 역할의 멤버인지 확인            | 280        |
| 23.12. ANSIBLE을 사용하여 호스트 그룹이 IDM RBAC 역할의 멤버인지 확인         | 282        |
| <b>24장. ANSIBLE 플레이북을 사용하여 RBAC 권한 관리</b>                 | <b>285</b> |
| 24.1. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한이 있는지 확인            | 285        |

|                                                       |            |
|-------------------------------------------------------|------------|
| 24.2. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한에 멤버 권한이 있는지 확인 | 287        |
| 24.3. ANSIBLE을 사용하여 IDM RBAC 권한에 권한이 포함되어 있지 않은지 확인   | 290        |
| 24.4. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한 이름 변경          | 292        |
| 24.5. ANSIBLE을 사용하여 IDM RBAC 권한이 없는지 확인               | 294        |
| 24.6. 추가 리소스                                          | 296        |
| <b>25장. ANSIBLE 플레이북을 사용하여 IDM에서 RBAC 권한 관리</b>       | <b>297</b> |
| 25.1. ANSIBLE을 사용하여 RBAC 권한이 있는지 확인                   | 297        |
| 25.2. ANSIBLE을 사용하여 특성이 있는 RBAC 권한이 있는지 확인합니다.        | 300        |
| 25.3. ANSIBLE을 사용하여 RBAC 권한이 없는지 확인                   | 303        |
| 25.4. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버인지 확인          | 304        |
| 25.5. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버가 아닌지 확인       | 306        |
| 25.6. ANSIBLE을 사용하여 IDM RBAC 권한 이름 변경                 | 309        |
| 25.7. 추가 리소스                                          | 310        |
| <b>26장. ID 보기를 사용하여 IDM 클라이언트의 사용자 속성 값 덮어쓰기</b>      | <b>312</b> |
| 26.1. ID 보기                                           | 312        |
| 26.2. SSSD 성능에 대한 ID 보기의 부정적인 영향                      | 314        |
| 26.3. ID 보기의 속성을 재정의할 수 있습니다.                         | 314        |
| 26.4. ID 보기 명령에 대한 도움말 가져오기                           | 315        |
| 26.5. ID 보기를 사용하여 특정 호스트에서 IDM 사용자의 로그인 이름 덮어쓰기       | 316        |
| 26.6. IDM ID 보기 수정                                    | 319        |
| 26.7. IDM 클라이언트의 IDM 사용자 홈 디렉터리를 덮어쓰는 ID 보기 추가        | 321        |
| 26.8. IDM 호스트 그룹에 ID 보기 적용                            | 324        |
| 26.9. NIS 도메인을 IDENTITY MANAGEMENT로 마이그레이션            | 327        |
| <b>27장. ACTIVE DIRECTORY 사용자를 위한 ID 보기 사용</b>         | <b>329</b> |
| 27.1. 기본 신뢰 보기의 작동 방식                                 | 329        |
| 27.2. 기본 신뢰 뷰를 수정하여 AD 사용자의 글로벌 속성 정의                 | 330        |
| 27.3. ID 보기를 사용하여 IDM 클라이언트의 AD 사용자의 기본 신뢰 보기 속성 덮어쓰기 | 331        |
| 27.4. IDM 호스트 그룹에 ID 보기 적용                            | 333        |
| <b>28장. 수동으로 ID 범위 조정</b>                             | <b>337</b> |
| 28.1. ID 범위                                           | 337        |
| 28.2. 자동 ID 범위 할당                                     | 338        |
| 28.3. 서버 설치 중 IDM ID 범위 수동 할당                         | 338        |
| 28.4. 새 IDM ID 범위 추가                                  | 339        |
| 28.5. IDM ID 범위에서 보안 및 상대 식별자의 역할                     | 341        |
| 28.6. AD에 대한 신뢰를 제거한 후 ID 범위 제거                       | 343        |
| 28.7. 현재 할당된 DNA ID 범위 표시                             | 344        |
| 28.8. 수동 ID 범위 할당                                     | 345        |
| 28.9. 수동으로 DNA ID 범위 할당                               | 346        |
| <b>29장. 하위 ID 범위 수동 관리</b>                            | <b>348</b> |
| 29.1. IDM CLI를 사용하여 SUBID 범위 생성                       | 348        |
| 29.2. IDM WEBUI 인터페이스를 사용하여 하위 ID 범위 생성               | 349        |
| 29.3. IDM CLI를 사용하여 기존 SUBID 범위 관리                    | 350        |
| 29.4. GETSUBID 명령을 사용하여 하위 ID 범위 나열                   | 351        |
| <b>30장. IDM CLI에서 호스트 관리</b>                          | <b>353</b> |
| 30.1. IDM의 호스트                                        | 353        |
| 30.2. 호스트 등록                                          | 354        |
| 30.3. 호스트 등록에 필요한 사용자 권한                              | 355        |
| 30.4. IDM 호스트 및 사용자의 등록 및 인증: 비교                      | 356        |
| 30.5. 호스트 작업                                          | 358        |

|                                                        |            |
|--------------------------------------------------------|------------|
| 30.6. IDM LDAP의 호스트 항목                                 | 360        |
| 30.7. IDM CLI에서 IDM 호스트 항목 추가                          | 361        |
| 30.8. IDM CLI에서 호스트 항목 삭제                              | 362        |
| 30.9. IDENTITY MANAGEMENT 클라이언트 다시 등록                  | 363        |
| 30.9.1. IdM의 클라이언트 re-enrollment                       | 363        |
| 30.9.2. 사용자 인증 정보를 사용하여 클라이언트 등록: 대화형 재등록              | 364        |
| 30.9.3. 클라이언트 키를 사용하여 클라이언트 인증: 비대화형 재등록               | 364        |
| 30.9.4. 설치 후 Identity Management 클라이언트 테스트             | 365        |
| 30.10. ID 관리 클라이언트 시스템 이름 변경                           | 365        |
| 30.10.1. IdM 클라이언트 이름 변경 준비                            | 366        |
| 30.10.2. ID 관리 클라이언트 설치 제거                             | 367        |
| 30.10.3. 호스트 시스템 이름 변경                                 | 368        |
| 30.10.4. 서비스 추가, 인증서 재 생성 및 호스트 그룹 다시 추가               | 368        |
| 30.11. 호스트 항목 비활성화 및 다시 활성화                            | 369        |
| 30.11.1. 호스트 비활성화                                      | 369        |
| 30.11.2. 호스트 다시 활성화                                    | 369        |
| <b>31장. IDM 웹 UI에서 호스트 항목 추가</b>                       | <b>371</b> |
| 31.1. IDM의 호스트                                         | 371        |
| 31.2. 호스트 등록                                           | 372        |
| 31.3. 호스트 등록에 필요한 사용자 권한                               | 372        |
| 31.4. IDM 호스트 및 사용자의 등록 및 인증: 비교                       | 373        |
| 31.5. IDM LDAP의 호스트 항목                                 | 375        |
| 31.6. 웹 UI에서 호스트 항목 추가                                 | 377        |
| <b>32장. ANSIBLE 플레이북을 사용하여 호스트 관리</b>                  | <b>380</b> |
| 32.1. ANSIBLE 플레이북을 사용하여 FQDN으로 IDM 호스트 항목이 있는지 확인     | 380        |
| 32.2. ANSIBLE 플레이북을 사용하여 DNS 정보로 IDM 호스트 항목이 있는지 확인    | 382        |
| 32.3. ANSIBLE 플레이북을 사용하여 임의의 암호로 여러 IDM 호스트 항목이 있는지 확인 | 384        |
| 32.4. ANSIBLE 플레이북을 사용하여 여러 IP 주소로 IDM 호스트 항목이 있는지 확인  | 386        |
| 32.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 항목이 없는지 확인            | 389        |
| 32.6. 추가 리소스                                           | 390        |
| <b>33장. IDM CLI를 사용하여 호스트 그룹 관리</b>                    | <b>391</b> |
| 33.1. IDM의 호스트 그룹                                      | 391        |
| 33.2. CLI를 사용하여 IDM 호스트 그룹 보기                          | 392        |
| 33.3. CLI를 사용하여 IDM 호스트 그룹 생성                          | 393        |
| 33.4. CLI를 사용하여 IDM 호스트 그룹 삭제                          | 393        |
| 33.5. CLI를 사용하여 IDM 호스트 그룹 멤버 추가                       | 394        |
| 33.6. CLI를 사용하여 IDM 호스트 그룹 멤버 제거                       | 395        |
| 33.7. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가                   | 397        |
| 33.8. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거                   | 399        |
| <b>34장. IDM 웹 UI를 사용하여 호스트 그룹 관리</b>                   | <b>401</b> |
| 34.1. IDM의 호스트 그룹                                      | 401        |
| 34.2. IDM 웹 UI에서 호스트 그룹 보기                             | 402        |
| 34.3. IDM 웹 UI에서 호스트 그룹 생성                             | 404        |
| 34.4. IDM 웹 UI에서 호스트 그룹 삭제                             | 404        |
| 34.5. IDM 웹 UI에 호스트 그룹 멤버 추가                           | 405        |
| 34.6. IDM 웹 UI에서 호스트 그룹 멤버 제거                          | 406        |
| 34.7. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가                  | 407        |
| 34.8. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거                  | 409        |
| <b>35장. ANSIBLE 플레이북을 사용하여 호스트 그룹 관리</b>               | <b>412</b> |

|                                                              |            |
|--------------------------------------------------------------|------------|
| 35.1. IDM의 호스트 그룹                                            | 412        |
| 35.2. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹이 있는지 확인                  | 413        |
| 35.3. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 호스트가 있는지 확인             | 415        |
| 35.4. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹 중첩                       | 417        |
| 35.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 멤버 관리자가 있는지 확인          | 418        |
| 35.6. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 호스트가 없는지 확인             | 420        |
| 35.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 중첩 호스트 그룹이 없는지 확인      | 422        |
| 35.8. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹이 없는지 확인                  | 424        |
| 35.9. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 멤버 관리자가 없는지 확인         | 426        |
| <b>36장. ANSIBLE 플레이북을 사용하여 IDM에 호스트 기반 액세스 제어 규칙이 있는지 확인</b> | <b>428</b> |
| 36.1. IDM의 호스트 기반 액세스 제어 규칙                                  | 428        |
| 36.2. ANSIBLE 플레이북을 사용하여 IDM에 HBAC 규칙이 있는지 확인                | 428        |
| <b>37장. 짧은 AD 사용자 이름을 확인하도록 도메인 확인 순서 구성</b>                 | <b>431</b> |
| 37.1. 도메인 확인 순서의 작동 방식                                       | 431        |
| 37.2. IDM 서버에서 글로벌 도메인 확인 순서 설정                              | 432        |
| 37.3. IDM 서버에서 ID 보기의 도메인 확인 순서 설정                           | 433        |
| 37.4. IDM 클라이언트의 SSSD에서 도메인 확인 순서 설정                         | 435        |
| 37.5. 추가 리소스                                                 | 436        |
| <b>38장. IDM에서 AD 사용자 계정 이름을 사용하여 인증 활성화</b>                  | <b>437</b> |
| 38.1. IDM에서 신뢰할 수 있는 AD FOREST의 사용자 보안 주체 이름                 | 437        |
| 38.2. IDM에서 AD UPNS가 최신 버전인지 확인                              | 438        |
| 38.3. AD UPN 인증 문제에 대한 문제 해결 데이터 수집                          | 439        |
| <b>39장. AD 사용자가 IDM을 관리할 수 있도록 활성화</b>                       | <b>441</b> |
| 39.1. AD 사용자의 ID 덮어쓰기                                        | 441        |
| 39.2. ID 덮어쓰기를 사용하여 AD 사용자가 IDM 관리 가능                        | 441        |
| 39.3. AD 사용자로 IDM CLI 관리                                     | 442        |



## 보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

## RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 이를 개선할 수 있는지 알려 주십시오.

- 특정 문구에 대한 간단한 의견 작성 방법은 다음과 같습니다.
  1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **피드백** 버튼이 있는지 확인합니다.
  2. 마우스 커서를 사용하여 주석 처리하려는 텍스트 부분을 강조 표시합니다.
  3. 강조 표시된 텍스트 아래에 표시되는 **피드백 추가** 팝업을 클릭합니다.
  4. 표시된 지침을 따릅니다.
- Bugzilla를 통해 피드백을 제출하려면 새 티켓을 생성합니다.
  1. [Bugzilla](#) 웹 사이트로 이동합니다.
  2. 구성 요소로 **문서**를 사용합니다.
  3. **설명** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
  4. **버그 제출**을 클릭합니다.

## 1장. IDM 명령줄 유틸리티 소개

다음 섹션에서는 IdM(Identity Management) 명령줄 유틸리티를 사용하는 기본 사항에 대해 설명합니다.

### 사전 요구 사항

- 설치 및 액세스할 수 있는 IdM 서버.  
자세한 내용은 [Identity Management 설치](#)를 참조하십시오.
- IPA 명령줄 인터페이스를 사용하려면 유효한 Kerberos 티켓으로 IdM을 인증합니다.

### 1.1. IPA 명령줄 인터페이스란 무엇입니까?

IPA 명령줄 인터페이스(CLI)는 IdM(Identity Management) 관리를 위한 기본 명령줄 인터페이스입니다.

IdM을 관리하는 다양한 하위 명령(예: **ipa user-add** 명령)을 지원하여 새 사용자를 추가합니다.

IPA CLI를 사용하면 다음을 수행할 수 있습니다.

- 네트워크에서 사용자, 그룹, 호스트 및 기타 오브젝트를 추가, 관리 또는 제거합니다.
- 인증서를 관리합니다.
- 검색 항목.
- 오브젝트를 표시하고 나열합니다.
- 액세스 권한을 설정합니다.
- 올바른 명령 구문에 대한 도움말을 가져옵니다.

### 1.2. IPA 도움말이란 무엇입니까?

IPA 도움말은 IdM 서버를 위한 기본 제공 문서 시스템입니다.

IPA 명령줄 인터페이스(CLI)는 로드된 IdM 플러그인 모듈의 사용 가능한 도움말 주제를 생성합니다. IPA 도움말 유틸리티를 사용하려면 다음이 필요합니다.

- IdM 서버가 설치되어 실행 중이어야 합니다.
- 유효한 Kerberos 티켓을 사용하여 인증합니다.

옵션 없이 **ipa help** 명령을 입력하면 기본 도움말 사용과 가장 일반적인 명령 예제에 대한 정보가 표시됩니다.

다음 옵션을 다양한 **ipa** 도움말 사용 사례에 사용할 수 있습니다.

**\$ ipa help [TOPIC | COMMAND | topics | commands]**

- **[]** "모든 매개변수는 선택 사항이며 **ipa** 도움말 만 쓸 수 있으며 명령을 실행할 수 있습니다.
- **|** 파이프 문자는 **또는** 을 의미합니다. 따라서 기본 **ipa help** 명령을 사용하여 **topic**, **COMMAND** 또는 **주제** 또는 명령을 지정할 수 있습니다.

- 주제: **ipa** 도움말 주제를 실행하여 **IPA** 도움말 (예: 사용자, 인증서, 서버 등)의 항목 목록을 표시할 수 있습니다.
- **top/Enical letters** 가 있는 **I/E**는 변수입니다. 따라서 특정 주제(예: **ipa help 사용자**)를 지정할 수 있습니다.
- **ipa help** 명령을 입력하여 **IPA** 도움말(예: **user-add,ca-enable,server-show** 등)의 명령 목록을 표시할 수 있습니다.
- 대/도: 대문자로 된 **Command MAND**는 변수입니다. 따라서 **ipa help user-add** 와 같은 특정 명령을 지정할 수 있습니다.

### 1.3. IPA 도움말 주제 사용

다음 절차에서는 명령줄 인터페이스에서 **IPA** 도움말을 사용하는 방법을 설명합니다.

#### 절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. **ipa** 도움말 주제를 입력하여 도움말에서 다루는 주제 목록을 표시합니다.

```
$ ipa help topics
```

3. 주제 중 하나를 선택하고 다음 패턴에 따라 명령을 만듭니다. **ipa help [topic\_name]**. **topic\_name** 문자열 대신 이전 단계에서 나열한 주제 중 하나를 추가합니다.

이 예제에서는 다음 주제를 사용합니다. **user**

```
$ ipa help user
```

4. **IPA** 도움말 출력이 너무 길어 전체 텍스트를 볼 수 없는 경우 다음 구문을 사용하십시오.

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

그런 다음 아래로 스크롤하여 전체 도움을 읽을 수 있습니다.

**IPA CLI**에는 사용자 항목에 대한 도움말 페이지가 표시됩니다. 개요를 읽은 후 주제 명령 작업을 위한 패턴의 많은 예제를 볼 수 있습니다.

#### 1.4. IPA 도움말 명령 사용

다음 절차에서는 명령줄 인터페이스에서 **IPA** 도움말 명령을 만드는 방법을 설명합니다.

##### 절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. **ipa help** 명령을 입력하여 도움말에서 다루는 명령 목록을 표시합니다.

```
$ ipa help commands
```

3. 명령 중 하나를 선택하고 다음 패턴에 따라 **help** 명령을 생성합니다. **ipa help <COMMAND>**. `<COMMAND>` 문자열 대신 이전 단계에서 나열한 명령 중 하나를 추가합니다.

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

##### 추가 리소스

- **ipa man** 페이지.

#### 1.5. IPA 명령 구조

**IPA CLI**는 다음 유형의 명령을 구분합니다.

- **IdM** 서버에서 기본 제공 명령 **tekton-databind built-in** 명령을 모두 사용할 수 있습니다.
- 플러그인 제공 명령

**IPA 명령의 구조를 사용하면 다양한 유형의 오브젝트를 관리할 수 있습니다. 예를 들어 다음과 같습니다.**

- 사용자,
- 호스트,
- **DNS 레코드,**
- 인증서,

그리고 더 많은

이러한 오브젝트 대부분에서 **IPA CLI**에는 다음과 같은 명령이 포함되어 있습니다.

- 추가(추가)
- 수정 (**mod**)
- 삭제(**del**)
- 검색 (찾기)
- 표시(표시)

명령에는 다음과 같은 구조가 있습니다.

**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 dns record-add, ipa dns records-mod, ipa dns records-del, ipa dns records-find, ipa dns records-show**

**ipa user-add [options]** 를 사용하여 사용자를 생성할 수 있습니다. 여기서 **[options]** 는 선택 사항입니다. **ipa user-add** 명령만 사용하면 스크립트에서 하나씩 세부 정보를 요청합니다.

기존 오브젝트를 변경하려면 오브젝트를 정의해야 합니다. 따라서 명령에는 오브젝트 **ipa user-mod USER\_NAME [options]** 도 포함됩니다.

## 1.6. IPA 명령을 사용하여 IDM에 사용자 계정 추가

다음 절차에서는 명령줄을 사용하여 **IdM(Identity Management)** 데이터베이스에 새 사용자를 추가하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 서버에 사용자 계정을 추가하려면 관리자 권한이 있어야 합니다.

절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. 새 사용자를 추가하려면 명령을 입력합니다.

```
$ ipa user-add
```

이 명령은 사용자 계정을 생성하는 데 필요한 기본 데이터를 제공하도록 요청하는 스크립트를 실행합니다.

3. **First name:** 필드에 새 사용자의 첫 번째 이름을 입력하고 **Enter** 키를 누릅니다.

4.

성: 필드에 새 사용자의 성을 입력하고 **Enter** 키를 누릅니다.

5.

**User login [suggested user name]:** 사용자 이름을 입력하거나 **Enter** 키를 눌러 제안된 사용자 이름을 수락합니다.

사용자 이름은 전체 **IdM** 데이터베이스에 대해 고유해야 합니다. 해당 사용자 이름이 이미 존재하기 때문에 오류가 발생하면 **ipa user-add** 명령을 사용하여 프로세스를 반복하고 다른 고유한 사용자 이름을 사용합니다.

사용자 이름을 추가한 후 사용자 계정이 **IdM** 데이터베이스에 추가되고 **IPA** 명령줄 인터페이스(**CLI**)는 다음 출력을 출력합니다.

```

-----
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

```

## 참고

기본적으로 사용자 암호는 사용자 계정에 설정되어 있지 않습니다. 사용자 계정을 생성하는 동안 암호를 추가하려면 다음 구문과 함께 **ipa user-add** 명령을 사용합니다.

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

IPA CLI에서 사용자 이름과 암호를 추가하거나 확인하라는 메시지가 표시됩니다.

사용자가 이미 생성된 경우 **ipa user-mod** 명령을 사용하여 암호를 추가할 수 있습니다.

## 추가 리소스

- 매개변수에 대한 자세한 내용을 보려면 **ipa help user-add** 명령을 실행합니다.

## 1.7. IPA 명령을 사용하여 IDM에서 사용자 계정 수정

각 사용자 계정에 대한 여러 매개 변수를 변경할 수 있습니다. 예를 들어 사용자에게 새 암호를 추가할 수 있습니다.

기본 명령 구문은 변경을 수행할 기존 사용자 계정을 정의해야 하므로 **user-add** 구문과 다릅니다(예: 암호 추가).

## 사전 요구 사항

- 사용자 계정을 수정하려면 관리자 권한이 있어야 합니다.

## 절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. **ipa user-mod** 명령을 입력하고 수정할 사용자를 지정하고, 암호를 추가하기 위해 **--password** 와 같은 옵션을 지정합니다.

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

이 명령은 새 암호를 추가할 수 있는 스크립트를 실행합니다.

3.

새 암호를 입력하고 **Enter** 키를 누릅니다.

IPA CLI는 다음 출력을 출력합니다.

```
-----
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
```

이제 계정에 사용자 암호가 설정되어 사용자가 **IdM**에 로그인할 수 있습니다.

## 추가 리소스

- 매개변수에 대한 자세한 내용은 **ipa help user-mod** 명령을 실행합니다.

## 1.8. IDM 유틸리티에 값 목록을 제공하는 방법

**IdM(Identity Management)**은 목록에 다중 값 특성 값을 저장합니다.

**IdM**은 다음과 같은 다중 값 목록을 제공하는 방법을 지원합니다.

- 동일한 명령 호출 내에서 동일한 명령줄 인수를 여러 번 사용합니다.

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

•

또는 목록을 중괄호로 묶을 수 있습니다. 이 경우 셸이 확장을 수행합니다.

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

위의 예제에서는 오브젝트에 권한을 추가하는 **permission-add** 명령을 보여줍니다. 이 예제에서는 개체를 언급하지 않습니다. ... 대신 권한을 추가하려는 오브젝트를 추가해야 합니다.

명령줄에서 이러한 다중 값 속성을 업데이트하면 **IdM**에서 이전 값 목록을 새 목록으로 완전히 덮어씁니다. 따라서 다중 값 특성을 업데이트할 때 추가하려는 단일 값이 아닌 전체 새 목록을 지정해야 합니다.

예를 들어 위의 명령에서 권한 목록에 읽기, 쓰기 및 삭제가 포함됩니다. **permission-mod** 명령으로 목록을 업데이트하려면 모든 값을 추가해야 합니다. 그렇지 않으면 언급되지 않은 값이 삭제됩니다.

예 1: **ipa permission-mod** 명령은 이전에 추가된 모든 권한을 업데이트합니다.

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

또는

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

예 2 - **ipa permission-mod** 명령은 명령에 포함되지 않기 때문에 **--right=delete** 인수를 삭제합니다.

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

또는

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

## 1.9. IDM 유틸리티와 특수 문자를 사용하는 방법

**ipa** 명령에 특수 문자가 포함된 명령줄 인수를 전달할 때 이러한 문자를 백슬래시(\)로 이스케이프합니다. 예를 들어, 일반적인 특수 문자에는 각도 대괄호(< 및 >), 앰퍼샌드(&), 별표(\*) 또는 수직 표시줄(|)이 포함됩니다.

예를 들어 별표 (\*) 를 이스케이프하려면 다음을 수행합니다.

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

셸에서 이러한 문자를 올바르게 구문 분석할 수 없기 때문에 이스케이프되지 않은 특수 문자가 포함된 명령은 예상대로 작동하지 않습니다.

## 2장. 명령줄을 사용하여 사용자 계정 관리

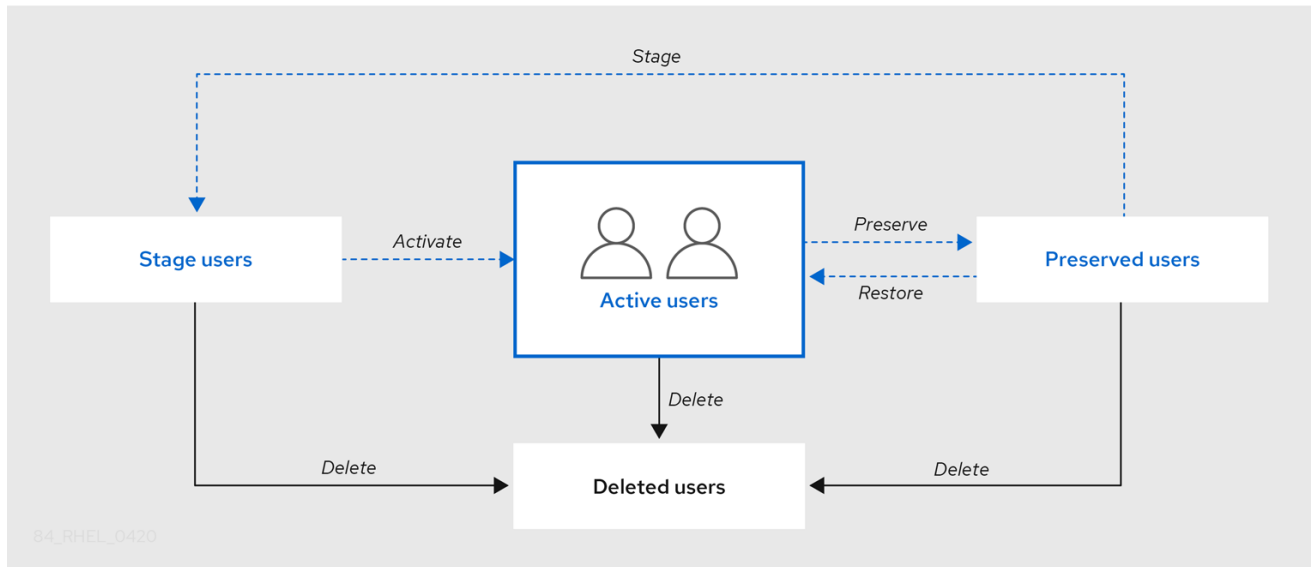
이 장에서는 **IdM(Identity Management)**의 사용자 라이프사이클에 대한 기본 설명이 포함되어 있습니다. 다음 섹션에서는 다음을 수행하는 방법을 보여줍니다.

- 사용자 계정 생성
- 단계적 사용자 계정 활성화
- 사용자 계정 보존
- 활성화, 스테이징 또는 보존된 사용자 계정 삭제
- 보존된 사용자 계정 복원

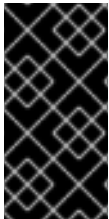
### 2.1. 사용자 라이프 사이클

**IdM(Identity Management)**은 세 가지 사용자 계정 상태를 지원합니다.

- 단계 사용자는 인증할 수 없습니다. 이는 초기 상태입니다. 활성화 사용자에게 필요한 일부 사용자 계정 속성은 설정할 수 없습니다(예: 그룹 멤버십).
- 활성화 사용자는 인증할 수 있습니다. 필요한 모든 사용자 계정 속성은 이 상태로 설정해야 합니다.
- 보존 사용자는 비활성 상태로 간주되는 이전 활성화 사용자이며 **IdM**에 인증할 수 없습니다. 보존 사용자는 활성화 사용자로 가지고 있는 대부분의 계정 속성을 유지하지만 사용자 그룹의 일부가 아닙니다.



IdM 데이터베이스에서 사용자 항목을 영구적으로 삭제할 수 있습니다.



#### 중요

삭제된 사용자 계정은 복원할 수 없습니다. 사용자 계정을 삭제하면 해당 계정과 연결된 모든 정보가 영구적으로 손실됩니다.

새 관리자는 기본 **admin** 사용자와 같은 관리자 권한이 있는 사용자만 만들 수 있습니다. 실수로 모든 관리자 계정을 삭제한 경우 **Directory Manager**는 **Directory Server**에서 수동으로 새 관리자를 생성해야 합니다.



#### 주의

**admin** 사용자를 삭제하지 마십시오. **admin**은 IdM에 필요한 사전 정의된 사용자이므로 이 작업으로 인해 특정 명령에 문제가 발생합니다. 대체 **admin** 사용자를 정의하고 사용하려면 하나 이상의 다른 사용자에게 관리자 권한을 부여한 후 **ipa user-disable admin** 사용자로 사전 정의된 **admin** 사용자를 비활성화하십시오.



### 주의

로컬 사용자를 **IdM**에 추가하지 마십시오. **NSS(Name Service Switch)**는 로컬 사용자 및 그룹을 확인하기 전에 항상 **IdM** 사용자 및 그룹을 확인합니다. 즉, 예를 들어 **IdM** 그룹 멤버십이 로컬 사용자에게는 작동하지 않습니다.

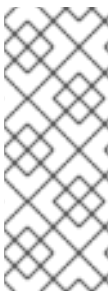
## 2.2. 명령줄을 사용하여 사용자 추가

다음과 같이 사용자를 추가할 수 있습니다.

- 사용자가 적극적으로 사용할 수 있는 활성 사용자 계정입니다.
- 해결 방법: 사용자는 이러한 계정을 사용할 수 없습니다. 새 사용자 계정을 준비하려는 경우 사용하십시오. 사용자가 계정을 사용할 준비가 되면 활성화할 수 있습니다.

다음 절차에서는 **ipa user-add** 명령을 사용하여 활성 사용자를 **IdM** 서버에 추가하는 방법을 설명합니다.

마찬가지로 **ipa stageuser-add** 명령을 사용하여 스테이지 사용자 계정을 생성할 수 있습니다.



### 참고

**IdM**은 고유한 사용자 **ID(UID)**를 새 사용자 계정에 자동으로 할당합니다. 수동으로 이 작업을 수행할 수도 있지만 서버에서 **UID** 번호가 고유한지 여부를 확인하지 않습니다. 이로 인해 여러 사용자 항목이 동일한 **ID** 번호가 할당될 수 있습니다. 동일한 **UID**가 있는 여러 항목이 없도록 하는 것이 좋습니다.

### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 받습니다.

## 절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. 사용자 로그인, 사용자 이름, 성 및 선택적으로 이메일 주소를 추가할 수도 있습니다.

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

**IdM**은 다음 정규식으로 설명할 수 있는 사용자 이름을 지원합니다.

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



## 참고

후행 달러 기호(\$)로 끝나는 사용자 이름은 **Samba 3.x** 시스템 지원을 활성화하기 위해 지원됩니다.

대문자가 포함된 사용자 이름을 추가하면 **IdM**에서 이름을 저장할 때 소문자로 자동 변환합니다. 따라서 **IdM**은 로그인 시 항상 소문자로 사용자 이름을 입력해야 합니다. 또한 사용자 및 사용자 등 문자 **casing**에만 다른 사용자 이름을 추가할 수 없습니다.

사용자 이름의 기본 최대 길이는 **32**자입니다. 변경하려면 **ipa config-mod --maxusername** 명령을 사용합니다. 예를 들어 최대 사용자 이름 길이를 **64**자로 늘리려면 다음을 수행합니다.

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

**ipa user-add** 명령에 많은 매개 변수가 포함되어 있습니다. 모두 나열하려면 **ipa help** 명령을 사용합니다.

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

**ipa help** 명령에 대한 자세한 내용은 [IPA 도움말](#)을 참조하십시오.

모든 **IdM** 사용자 계정을 나열하여 새 사용자 계정이 생성되었는지 확인할 수 있습니다.

**\$ ipa user-find**

이 명령은 모든 사용자 계정과 세부 사항을 나열합니다.

**2.3. 명령줄을 사용하여 사용자 활성화**

사용자 계정을 스테이지에서 활성 상태로 이동하여 활성화하려면 **ipa stageuser-activate** 명령을 사용합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 받습니다.

절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. 다음 명령을 사용하여 사용자 계정을 활성화합니다.

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

모든 **IdM** 사용자 계정을 나열하여 새 사용자 계정이 생성되었는지 확인할 수 있습니다.

**\$ ipa user-find**

이 명령은 모든 사용자 계정과 세부 사항을 나열합니다.

**2.4. 명령줄을 사용하여 사용자 보존**

사용자 계정을 제거하려면 사용자 계정을 보존할 수 있지만 나중에 복원할 수 있는 옵션을 유지합니다.

사용자 계정을 보존하려면 **ipa user-del** 또는 **ipa stageuser-del** 명령과 함께 **--preserve** 옵션을 사용합니다.

#### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 받습니다.

#### 절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. 다음 명령을 사용하여 사용자 계정을 보존합니다.

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

```
Deleted user "user_login"
```



#### 참고

사용자 계정이 삭제되었음을 나타내는 출력에도 그대로 유지됩니다.

## 2.5. 명령줄을 사용하여 사용자 삭제

**IdM(Identity Management)**을 사용하면 사용자를 영구적으로 삭제할 수 있습니다. 다음을 삭제할 수 있습니다.

- 다음 명령을 사용하는 활성 사용자: **ipa user-del**
- 다음 명령을 사용하여 사용자를 스테이징합니다. **ipa stageuser-del**
- 다음 명령을 사용하여 보존된 사용자: **ipa user-del**

여러 사용자를 삭제할 때 **--continue** 옵션을 사용하여 오류에 관계없이 명령을 강제로 계속합니다. 명령이 완료되면 성공 및 실패한 작업에 대한 요약이 **stdout** 표준 출력 스트림에 출력됩니다.

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

**--continue** 를 사용하지 않는 경우 명령은 오류가 발생하여 중지되고 종료될 때까지 사용자 삭제를 진행합니다.

#### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 받습니다.

#### 절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. 다음 명령을 사용하여 사용자 계정을 삭제합니다.

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

사용자 계정이 **IdM**에서 영구적으로 삭제되었습니다.

## 2.6. 명령줄을 사용하여 사용자 복원

보존 사용자를 다음과 같이 복원할 수 있습니다.

- 활성 사용자: **ipa user-undel**
- 사용자 단계: **ipa user-stage**

사용자 계정을 복원해도 계정의 이전 속성이 모두 복원되지는 않습니다. 예를 들어 사용자 암호는 복원되지 않으며 다시 설정해야 합니다.

#### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 받습니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#)을 참조하십시오.

#### 절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. 다음 명령을 사용하여 사용자 계정을 활성화합니다.

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

또는 사용자 계정을 **staged**로 복원할 수 있습니다.

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

#### 검증 단계

- 모든 **IdM** 사용자 계정을 나열하여 새 사용자 계정이 생성되었는지 확인할 수 있습니다.

```
$ ipa user-find
```

이 명령은 모든 사용자 계정과 세부 사항을 나열합니다.

### 3장. IDM 웹 UI를 사용하여 사용자 계정 관리

**IdM(Identity Management)**은 다양한 사용자 작업 상황을 관리하는 데 도움이 될 수 있는 **여러 단계를** 제공합니다.

#### 사용자 계정 생성

직원이 회사에서 경력을 시작하기 전에 **스테이지 사용자 계정을 만들고** 직원이 사무실에 표시되고 계정을 활성화하려면 사전에 준비하십시오.

이 단계를 생략하고 활성 사용자 계정을 직접 생성할 수 있습니다. 절차는 단계적 사용자 계정을 생성하는 것과 유사합니다.

#### 사용자 계정 활성화

직원의 첫 번째 작업일을 **계정**을 활성화합니다.

#### 사용자 계정 비활성화

사용자가 몇 개월 동안 부모휴대에 갈 경우 **계정을 일시적으로 비활성화해야 합니다.**

#### 사용자 계정 활성화

사용자가 반환하면 **계정을 다시 활성화해야 합니다.**

#### 사용자 계정 유지

사용자가 퇴사하려는 경우 일정 시간 후에 다시 돌아올 수 있으므로 **계정을 복원 할 수있는 가능성을 삭제해야 합니다.**

#### 사용자 계정 복원

**2년 후** 사용자는 다시 돌아가서 **보존된 계정을 복원해야 합니다.**

#### 사용자 계정 삭제

직원이 차감된 경우 백업없이 **계정을 삭제합니다.**

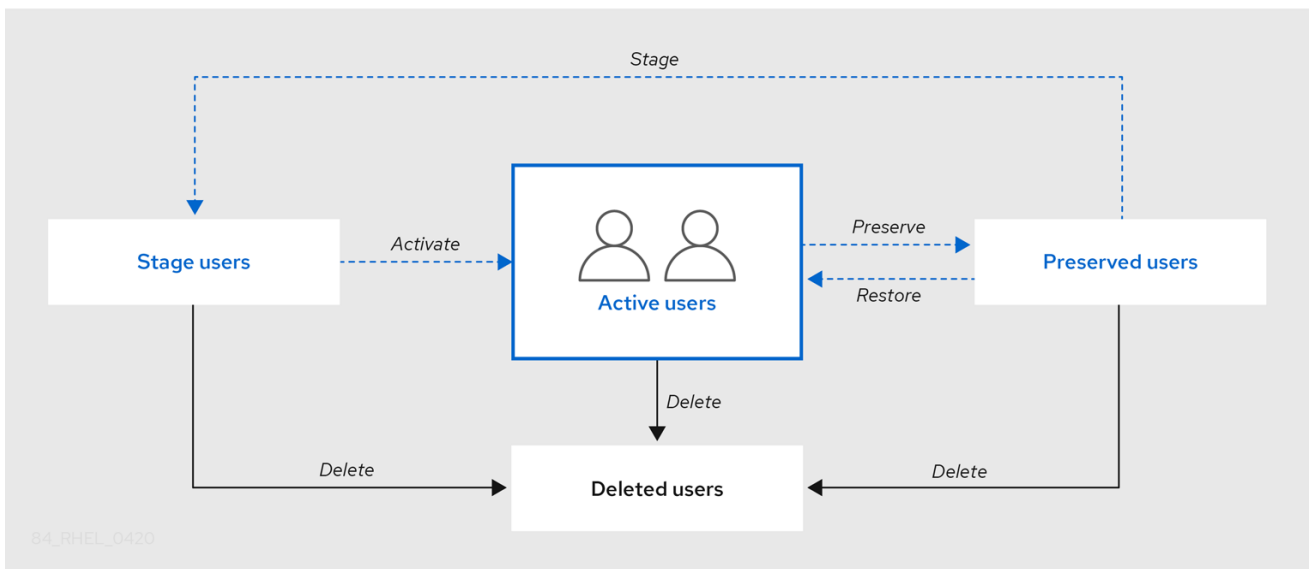
#### 3.1. 사용자 라이프 사이클

**IdM(Identity Management)**은 세 가지 사용자 계정 상태를 지원합니다.

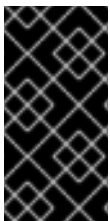
-

단계 사용자는 인증할 수 없습니다. 이는 초기 상태입니다. 활성 사용자에게 필요한 일부 사용자 계정 속성은 설정할 수 없습니다(예: 그룹 멤버십).

- 활성 사용자는 인증할 수 있습니다. 필요한 모든 사용자 계정 속성은 이 상태로 설정해야 합니다.
- 보존 사용자는 비활성 상태로 간주되는 이전 활성 사용자이며 IdM에 인증할 수 없습니다. 보존 사용자는 활성 사용자로 가지고 있는 대부분의 계정 속성을 유지하지만 사용자 그룹의 일부가 아닙니다.



IdM 데이터베이스에서 사용자 항목을 영구적으로 삭제할 수 있습니다.



#### 중요

삭제된 사용자 계정은 복원할 수 없습니다. 사용자 계정을 삭제하면 해당 계정과 연결된 모든 정보가 영구적으로 손실됩니다.

새 관리자는 기본 **admin** 사용자와 같은 관리자 권한이 있는 사용자만 만들 수 있습니다. 실수로 모든 관리자 계정을 삭제한 경우 **Directory Manager**는 **Directory Server**에서 수동으로 새 관리자를 생성해야 합니다.



## 주의

**admin** 사용자를 삭제하지 마십시오. **admin**은 **IdM**에 필요한 사전 정의의 사용자이므로 이 작업으로 인해 특정 명령에 문제가 발생합니다. 대체 **admin** 사용자를 정의하고 사용하려면 하나 이상의 다른 사용자에게 관리자 권한을 부여한 후 **ipa user-disable admin** 사용자로 사전 정의된 **admin** 사용자를 비활성화하십시오.



## 주의

로컬 사용자를 **IdM**에 추가하지 마십시오. **NSS(Name Service Switch)**는 로컬 사용자 및 그룹을 확인하기 전에 항상 **IdM** 사용자 및 그룹을 확인합니다. 즉, 예를 들어 **IdM** 그룹 멤버십이 로컬 사용자에게는 작동하지 않습니다.

### 3.2. 웹 UI에서 사용자 추가

일반적으로 새 직원이 작업을 시작하기 전에 새 사용자 계정을 만들어야 합니다. 이러한 스테이지 계정에 액세스할 수 없으며 나중에 활성화해야 합니다.



## 참고

또는 활성 사용자 계정을 직접 만들 수도 있습니다. 활성 사용자를 추가하려면 아래 절차에 따라 활성 사용자 탭에 사용자 계정을 추가하십시오.

#### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

#### 절차

1. **IdM 웹 UI**에 로그인합니다.

2.

사용자 → 단계 사용자 탭으로 이동합니다.

또는 사용자 → 활성 사용자에서 사용자 계정을 추가할 수 있지만 계정에 사용자 그룹을 추가할 수 없습니다.

3.

+ 추가 아이콘을 클릭합니다.

4.

**Add stage user dialog** 상자에 새 사용자의 성과 성을 입력합니다.

5.

[선택 사항] 사용자 로그인 필드에 로그인 이름을 추가합니다.

비워 두는 경우 **IdM** 서버는 다음 패턴에 로그인 이름을 생성합니다. 첫 번째 문자와 성의 첫 글자입니다. 전체 로그인 이름은 최대 **32**자까지 포함할 수 있습니다.

6.

[선택 사항] **GID** 드롭다운 메뉴에서 사용자를 포함해야 하는 그룹을 선택합니다.

7.

[선택 사항] 암호 필드에 암호를 입력하고 암호를 입력하여 둘 다 일치하는지 확인합니다.

8.

추가 버튼을 클릭합니다.

Add stage user

User login

First name \*

Example

Last name \*

User

Class

New Password

••••••••

Verify Password

••••••••

\* Required field

Add

Add and Add Another

Add and Edit

Cancel

이 시점에서 **Stage Users** 테이블에서 사용자 계정을 볼 수 있습니다.

RED HAT IDENTITY MANAGEMENT

Administrator

Identity

Policy

Authentication

Network Services

IPA Server

Users

Hosts

Services

Groups

ID Views

Automember

User categories

Active users

Stage users

Preserved users

Stage Users

Search

Refresh

Delete

+ Add

Activate

|                          | User login | First name | Last name | UID | Email address         | Telephone Number | Job Title |
|--------------------------|------------|------------|-----------|-----|-----------------------|------------------|-----------|
| <input type="checkbox"/> | euser      | Example    | User      | -1  | euser@idm.example.com |                  |           |

Showing 1 to 1 of 1 entries.



## 참고

사용자 이름을 클릭하면 전화 번호, 주소 또는 작업장 추가와 같은 고급 설정을 편집할 수 있습니다.

### 3.3. IDM 웹 UI에서 스테이징 사용자 활성화

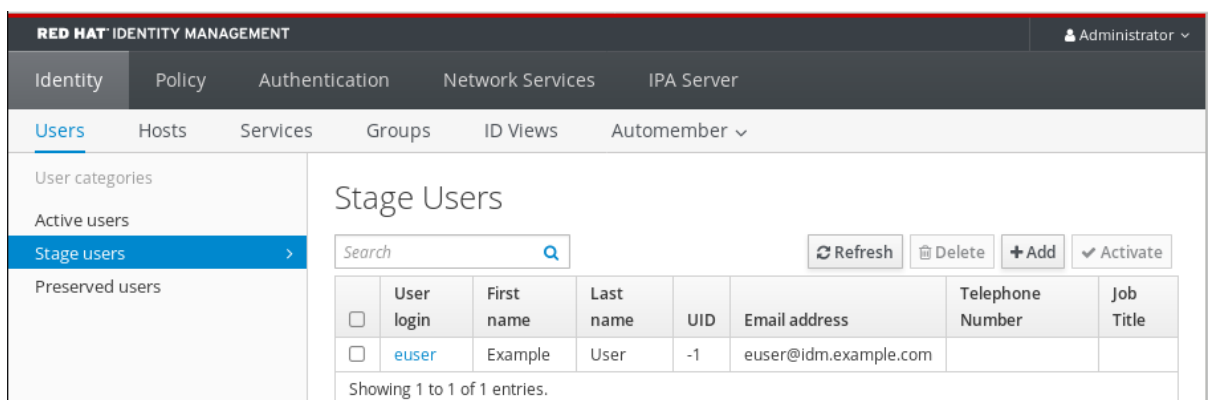
사용자가 IdM에 로그인하려면 사용자를 IdM 그룹에 추가하려면 스테이지 사용자 계정을 활성화해야 합니다. 이 섹션에서는 단계적 사용자 계정을 활성화하는 방법에 대해 설명합니다.

#### 사전 요구 사항

- IdM 웹 UI 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM에 준비된 사용자 계정이 한 개 이상 있습니다.

#### 절차

1. IdM 웹 UI에 로그인합니다.
2. 사용자 → 단계 사용자 탭으로 이동합니다.
3. 활성화할 사용자 계정의 확인란을 클릭합니다.
4. redfish 버튼을 클릭합니다.



5. 확인 대화 상자에서 확인 버튼을 클릭합니다.

활성화에 성공하면 IdM 웹 UI에 사용자가 활성화되었으며 사용자 계정이 활성 사용자로 이동되었다는

녹색 확인이 표시됩니다. 계정이 활성화되어 있으며 사용자는 **IdM** 도메인 및 **IdM 웹 UI**에 대해 인증할 수 있습니다. 처음 로그인할 때 사용자에게 암호를 변경하라는 메시지가 표시됩니다.

Active users

| <input type="text" value="Search"/> |             | <input type="button" value="Refresh"/> |               | <input type="button" value="Delete"/> |          | <input type="button" value="+ Add"/> |                  | <input type="button" value="- Disable"/> |  | <input type="button" value="✓ Enable"/> |  | <input type="button" value="Actions ▼"/> |  |
|-------------------------------------|-------------|----------------------------------------|---------------|---------------------------------------|----------|--------------------------------------|------------------|------------------------------------------|--|-----------------------------------------|--|------------------------------------------|--|
| <input type="checkbox"/>            | User login  | First name                             | Last name     | Status                                | UID      | Email address                        | Telephone Number | Job Title                                |  |                                         |  |                                          |  |
| <input type="checkbox"/>            | admin       |                                        | Administrator | ✓ Enabled                             | 78000000 |                                      |                  |                                          |  |                                         |  |                                          |  |
| <input checked="" type="checkbox"/> | euser       | Example                                | User          | ✓ Enabled                             | 78000006 | euser@idm.example.com                |                  |                                          |  |                                         |  |                                          |  |
| <input type="checkbox"/>            | staged.user | Staged                                 | User          | ✓ Enabled                             | 78000008 | staged.user@idm.example.com          |                  |                                          |  |                                         |  |                                          |  |

Showing 1 to 3 of 3 entries.



#### 참고

이 단계에서는 활성 사용자 계정을 사용자 그룹에 추가할 수 있습니다.

### 3.4. 웹 UI에서 사용자 계정 비활성화

활성 사용자 계정을 비활성화할 수 있습니다. 사용자 계정을 비활성화하면 계정이 비활성화되므로 사용자 계정을 사용하여 **Kerberos**와 같은 **IdM** 서비스를 인증하거나 작업을 수행할 수 없습니다.

**IdM** 내에는 비활성화된 사용자 계정이 여전히 존재하며 관련 정보는 모두 변경되지 않은 상태로 유지됩니다. 보존된 사용자 계정과 달리 비활성화된 사용자 계정은 활성 상태로 유지되며 사용자 그룹의 멤버일 수 있습니다.



#### 참고

사용자 계정을 비활성화한 후 기존 연결은 사용자의 **Kerberos TGT** 및 기타 티켓이 만료될 때까지 유효합니다. 티켓이 만료되면 사용자는 이를 갱신할 수 없습니다.

#### 사전 요구 사항



**IdM 웹 UI** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

## 절차

1. **IdM 웹 UI에 로그인합니다.**
2. **사용자 → 활성 사용자 탭으로 이동합니다.**
3. **비활성화할 사용자 계정의 확인란을 클릭합니다.**
4. **Disable (비활성화) 버튼을 클릭합니다.**

Active users

Search

Refresh Delete Add Disable Enable Actions

| <input type="checkbox"/>            | User login     | First name | Last name     | Status    | UID      | Email address                  | Telephone Number | Job Title |
|-------------------------------------|----------------|------------|---------------|-----------|----------|--------------------------------|------------------|-----------|
| <input type="checkbox"/>            | admin          |            | Administrator | ✓ Enabled | 78000000 |                                |                  |           |
| <input checked="" type="checkbox"/> | euser          | Example    | User          | ✓ Enabled | 78000006 | euser@idm.example.com          |                  |           |
| <input type="checkbox"/>            | preserved.user | Preserved  | User          | ✓ Enabled | 78000009 | preserved.user@idm.example.com |                  |           |

Showing 1 to 3 of 3 entries.

5. **확인 대화 상자에서 확인 버튼을 클릭합니다.**

비활성화 프로시저가 성공한 경우 활성 **users** 테이블의 상태 열을 확인할 수 있습니다.

| <input type="checkbox"/> | User login     | First name | Last name     | Status     | UID      | Email address                  | Telephone Number |
|--------------------------|----------------|------------|---------------|------------|----------|--------------------------------|------------------|
| <input type="checkbox"/> | admin          |            | Administrator | ✓ Enabled  | 78000000 |                                |                  |
| <input type="checkbox"/> | euser          | Example    | User          | — Disabled | 78000006 | euser@idm.example.com          |                  |
| <input type="checkbox"/> | preserved.user | Preserved  | User          | ✓ Enabled  | 78000009 | preserved.user@idm.example.com |                  |

### 3.5. 웹 UI에서 사용자 계정 활성화

IdM을 사용하면 활성 사용자 계정을 비활성화할 수 있습니다. 사용자 계정을 활성화하면 비활성화된 계정이 활성화됩니다.

사전 요구 사항

- **IdM 웹 UI 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**

절차

1. **IdM 웹 UI에 로그인합니다.**
2. **사용자 → 활성 사용자 탭으로 이동합니다.**
3. **활성화할 사용자 계정의 확인란을 클릭합니다.**
4. **Enable 버튼을 클릭합니다.**

Active users

| Search <input type="text"/>         |                | Refresh    |               | Delete    | + Add    | <b>Disable</b>                 | <b>Enable</b>    | Actions ▾ |
|-------------------------------------|----------------|------------|---------------|-----------|----------|--------------------------------|------------------|-----------|
| <input type="checkbox"/>            | User login     | First name | Last name     | Status    | UID      | Email address                  | Telephone Number | Job Title |
| <input type="checkbox"/>            | admin          |            | Administrator | ✓ Enabled | 78000000 |                                |                  |           |
| <input checked="" type="checkbox"/> | euser          | Example    | User          | ✓ Enabled | 78000006 | euser@idm.example.com          |                  |           |
| <input type="checkbox"/>            | preserved.user | Preserved  | User          | ✓ Enabled | 78000009 | preserved.user@idm.example.com |                  |           |

Showing 1 to 3 of 3 entries.

5. **확인 대화 상자에서 확인 버튼을 클릭합니다.**

변경이 성공하면 **Active users** 테이블의 상태 열을 확인할 수 있습니다.

### 3.6. IDM 웹 UI에서 활성 사용자 유지

사용자 계정을 보존하면 활성 사용자 탭에서 계정을 제거하고 **IdM**에 이러한 계정을 유지할 수 있습니다.

직원이 퇴사하는 경우 사용자 계정을 보존합니다. 몇 주 또는 몇 달 동안 사용자 계정을 비활성화하려는 경우(예: 초급 종료) 계정을 비활성화합니다. 자세한 내용은 [웹 UI에서 사용자 계정 비활성화](#)를 참조하십시오. 보존된 계정이 활성화되어 있지 않으며 사용자는 이를 사용하여 내부 네트워크에 액세스할 수 없지만 계정은 모든 데이터와 함께 데이터베이스에 남아 있습니다.

복원된 계정을 활성 모드로 다시 이동할 수 있습니다.



참고

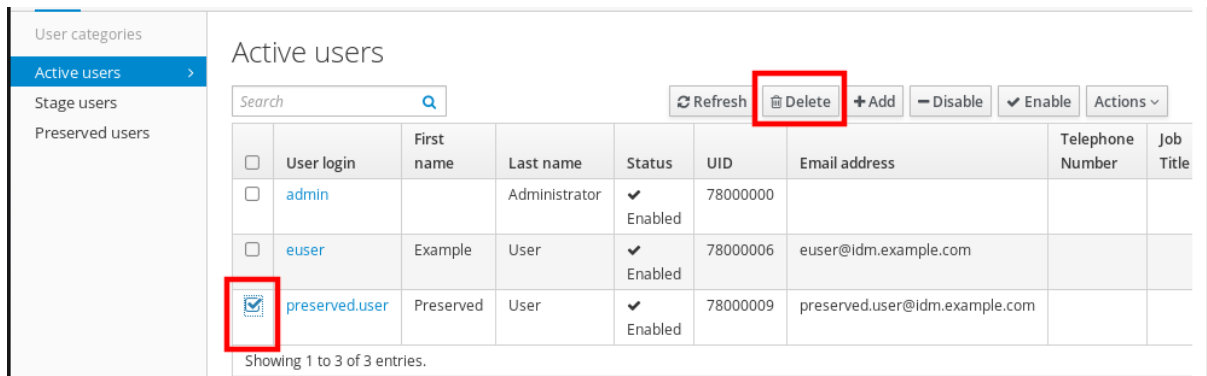
보존된 상태의 사용자 목록은 과거 사용자 계정의 기록을 제공할 수 있습니다.

사전 요구 사항

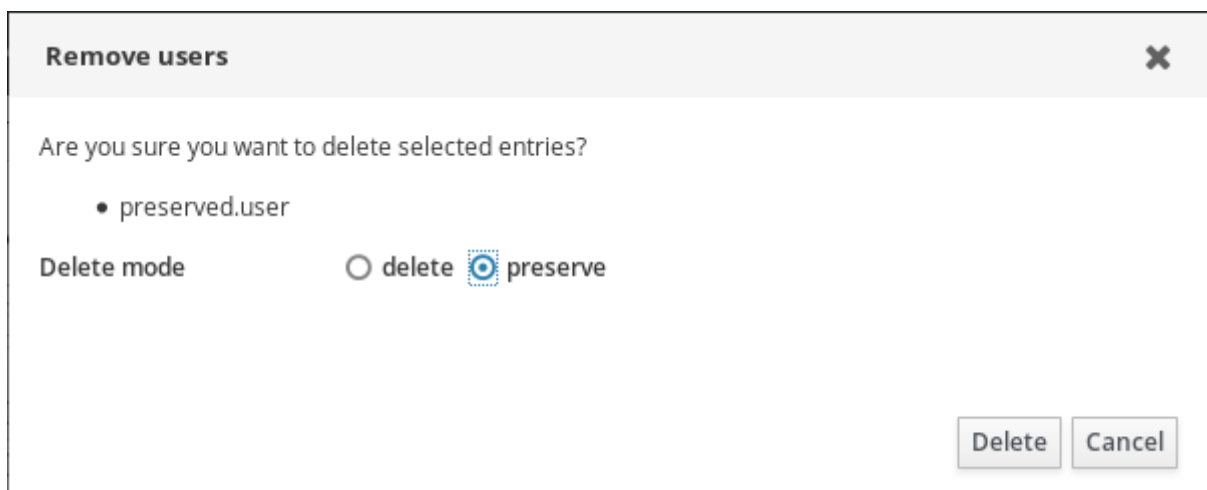
- **IdM(Identity Management) 웹 UI** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

절차

1. **IdM 웹 UI에 로그인합니다.**
2. 사용자 → 활성 사용자 탭으로 이동합니다.
3. 보존할 사용자 계정의 확인란을 클릭합니다.
4. **Delete** 버튼을 클릭합니다.



5. 사용자 제거 대화 상자에서 삭제 모드 라디오 버튼을 전환하여 보존 합니다.
6. **Delete** 버튼을 클릭합니다.



결과적으로 사용자 계정이 **Preserved** 사용자로 이동됩니다.

보존된 사용자를 복원해야 하는 경우 **IdM 웹 UI의 Restoring 사용자**를 참조하십시오.

### 3.7. IDM 웹 UI에서 사용자 복원

**IdM(Identity Management)**을 사용하면 보존된 사용자 계정을 활성 상태로 다시 복원할 수 있습니다.

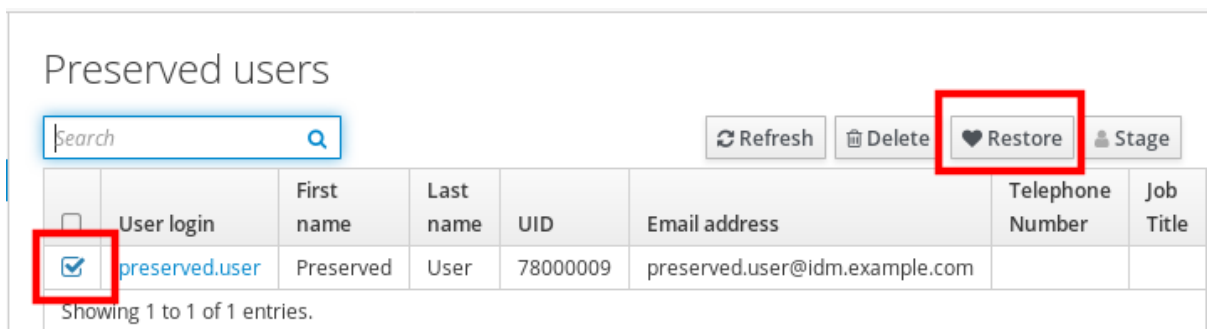
사전 요구 사항

•

IdM 웹 UI 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

#### 절차

1. IdM 웹 UI에 로그인합니다.
2. 사용자 → **Preserved** 사용자 탭으로 이동합니다.
3. 복원할 사용자 계정에서 확인란을 클릭합니다.
4. **Restore** 버튼을 클릭합니다.



5. 확인 대화 상자에서 확인 버튼을 클릭합니다.

IdM 웹 UI는 녹색 확인을 표시하고 사용자 계정을 활성 사용자 탭으로 이동합니다.

### 3.8. IDM 웹 UI에서 사용자 삭제

사용자 삭제는 되돌릴 수 없는 작업이므로 그룹 멤버십 및 암호를 포함하여 사용자 계정이 IdM 데이터베이스에서 영구적으로 삭제됩니다. 시스템 계정 및 홈 디렉터리와 같은 사용자의 외부 구성은 삭제되지 않지만 IdM을 통해 더 이상 액세스할 수 없습니다.

다음은 삭제할 수 있습니다.

- - 활성 사용자 **already** - **IdM 웹 UI**는 다음과 같은 옵션을 제공합니다.
  - - 사용자를 임시로 유지
    - 자세한 내용은 **IdM 웹 UI의 활성 사용자** 예약을 참조하십시오.
    - 영구적으로 삭제
- **Stage users:you can just delete stage users permanently...** 이 사용자는 단계의 사용자를 영구적으로 삭제할 수 있습니다.
- 보존된 사용자: 보존된 사용자를 영구적으로 삭제할 수 있습니다.

다음 절차에서는 활성 사용자 삭제를 설명합니다. 마찬가지로 다음에서 사용자 계정을 삭제할 수 있습니다.

- **Stage 사용자 탭**
- **Preserved 사용자 탭**

사전 요구 사항

- **IdM 웹 UI** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

절차

1. **IdM 웹 UI**에 로그인합니다.
2. 사용자 → 활성 사용자 탭으로 이동합니다.

또는 사용자 → 스테이지 사용자 또는 사용자 → **Preserved 사용자** 에서 사용자 계정을 삭제할 수 있습니다.

3.

**Delete** 아이콘을 클릭합니다.

4.

사용자 제거 대화 상자에서 삭제 모드 라디오 버튼을 전환하여 을 삭제합니다 .

5.

**Delete** 버튼을 클릭합니다.

사용자 계정이 **IdM**에서 영구적으로 삭제되었습니다.

## 4장. ANSIBLE 플레이북을 사용하여 사용자 계정 관리

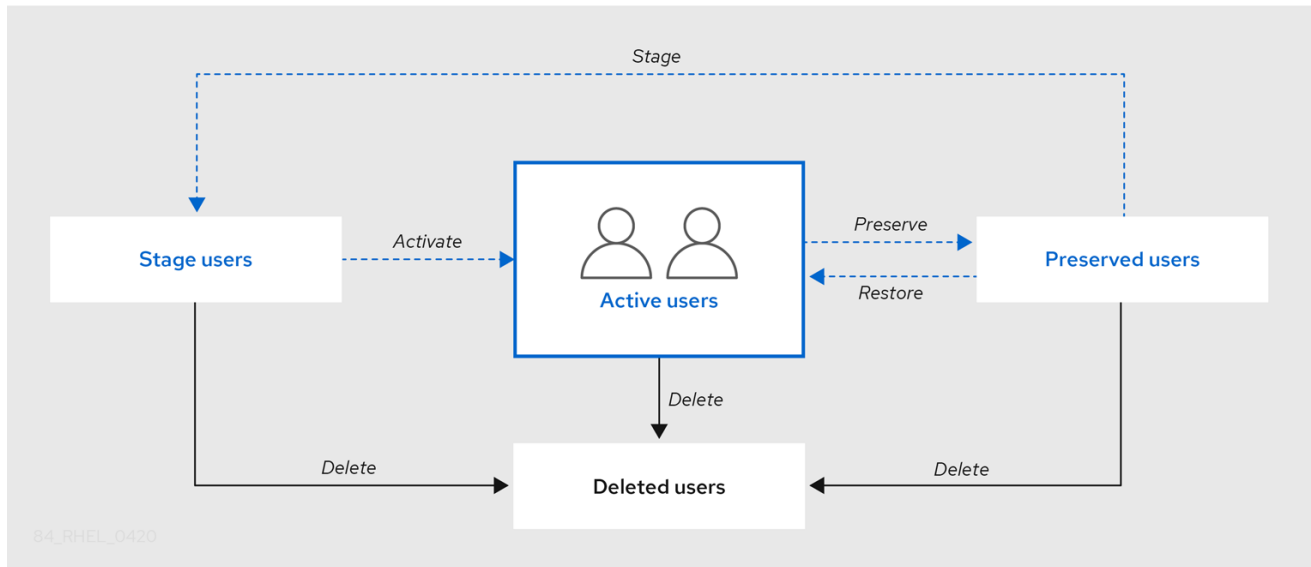
**Ansible** 플레이북을 사용하여 **IdM**에서 사용자를 관리할 수 있습니다. [사용자 라이프사이클](#) 을 제공한 후 이 장에서는 다음 작업에 **Ansible** 플레이북을 사용하는 방법을 설명합니다.

- **YML** 파일에 직접 나열된 **단일 사용자가** 있는지 확인합니다.
- **YML** 파일에 직접 나열된 **여러 사용자가** 있는지 확인합니다.
- **JSON** 파일에 나열된 **여러 사용자가 YML** 파일에서 참조되는지 확인합니다.
- **YML** 파일에 직접 나열된 **사용자가 없는지** 확인합니다.

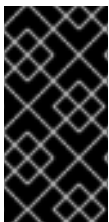
### 4.1. 사용자 라이프 사이클

**IdM(Identity Management)**은 세 가지 사용자 계정 상태를 지원합니다.

- **단계 사용자**는 인증할 수 없습니다. 이는 초기 상태입니다. 활성 사용자에게 필요한 일부 사용자 계정 속성은 설정할 수 없습니다(예: 그룹 멤버십).
- **활성 사용자**는 인증할 수 있습니다. 필요한 모든 사용자 계정 속성은 이 상태로 설정해야 합니다.
- **보존 사용자**는 비활성 상태로 간주되는 이전 활성 사용자이며 **IdM**에 인증할 수 없습니다. 보존 사용자는 활성 사용자로 가지고 있는 대부분의 계정 속성을 유지하지만 사용자 그룹의 일부가 아닙니다.



IdM 데이터베이스에서 사용자 항목을 영구적으로 삭제할 수 있습니다.



#### 중요

삭제된 사용자 계정은 복원할 수 없습니다. 사용자 계정을 삭제하면 해당 계정과 연결된 모든 정보가 영구적으로 손실됩니다.

새 관리자는 기본 **admin** 사용자와 같은 관리자 권한이 있는 사용자만 만들 수 있습니다. 실수로 모든 관리자 계정을 삭제한 경우 **Directory Manager**는 **Directory Server**에서 수동으로 새 관리자를 생성해야 합니다.



#### 주의

**admin** 사용자를 삭제하지 마십시오. **admin**은 IdM에 필요한 사전 정의된 사용자이므로 이 작업으로 인해 특정 명령에 문제가 발생합니다. 대체 **admin** 사용자를 정의하고 사용하려면 하나 이상의 다른 사용자에게 관리자 권한을 부여한 후 **ipa user-disable admin** 사용자로 사전 정의된 **admin** 사용자를 비활성화하십시오.



## 주의

로컬 사용자를 **IdM**에 추가하지 마십시오. **NSS(Name Service Switch)**는 로컬 사용자 및 그룹을 확인하기 전에 항상 **IdM** 사용자 및 그룹을 확인합니다. 즉, 예를 들어 **IdM** 그룹 멤버십이 로컬 사용자에게는 작동하지 않습니다.

## 4.2. ANSIBLE 플레이북을 사용하여 IDM 사용자가 있는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM**에 사용자가 있는지 확인하는 방법을 설명합니다.

### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **ansible-freeipa** 패키지는 **Ansible** 컨트롤러에 설치됩니다.

### 절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 **ipaserver** 를 이 파일에 정의합니다.

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

2. 다음을 보장하려는 **IdM**에 있는 사용자의 데이터를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/user/add-user.yml** 파일에서 예제를 복사하고 수정할 수 있습니다. 예를 들어 이름이 **idm\_user** 인 사용자를 생성하고 사용자 암호로 **Password123** 을 추가하려면 다음을 수행합니다.

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

  tasks:
    - name: Create user idm_user
      ipauser:
        ipaadmin_password: MySecret123
        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

```

다음 옵션을 사용하여 사용자를 추가해야 합니다.

- 이름: 로그인 이름
- First: 첫 번째 이름 문자열
- Last: 마지막 이름 문자열

사용 가능한 사용자 옵션의 전체 목록은 `/usr/share/doc/ansible-freeipa/README-user.md` Markdown 파일을 참조하십시오.



참고

`update_password: on_create` 옵션을 사용하는 경우 **Ansible**은 사용자를 생성할 때만 사용자 암호를 생성합니다. 사용자가 암호를 사용하여 이미 생성된 경우 **Ansible**에서 새 암호를 생성하지 않습니다.

3. 플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/add-IdM-user.yml

```

검증 단계

- `ipa user-show` 명령을 사용하여 IdM에 새 사용자 계정이 있는지 확인할 수 있습니다.
1. **admin**으로 **ipaserver**에 로그인합니다.

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

2. 관리자를 위한 **Kerberos** 티켓을 요청합니다.

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

3. *idm\_user*에 대한 정보 요청 :

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

*idm\_user* 라는 사용자는 **IdM**에 있습니다.

#### 4.3. ANSIBLE 플레이북을 사용하여 여러 **IDM** 사용자가 있는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM**에 여러 사용자가 있는지 확인하는 방법을 설명합니다.

##### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.

##### 절차

1. 인벤토리 파일(예: *inventory.file*)을 생성하고 **ipaserver**를 이 파일에 정의합니다.

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

2. **IdM**에 확인하려는 사용자의 데이터를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 이 단

계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml` 파일에서 예제를 복사하고 수정할 수 있습니다. 예를 들어 사용자 `idm_user_1`, `idm_user_2`, `idm_user_3` 을 만들고 `Password123` 을 `idm_user_1` 의 암호로 추가하려면 다음을 수행합니다.

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

  tasks:
  - name: Create user idm_users
    ipauser:
      ipaadmin_password: MySecret123
      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
```

#### 참고

`update_password: on_create` 옵션을 지정하지 않으면 **Ansible**은 플레이북이 실행될 때마다 사용자 암호를 다시 설정합니다. 즉, 사용자가 플레이북을 마지막으로 실행한 후 암호를 변경한 경우 **Ansible**은 암호를 다시 설정합니다.

3.

플레이북을 실행합니다.

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

검증 단계

- **ipa user-show** 명령을 사용하여 **IdM**에 사용자 계정이 있는지 확인할 수 있습니다.

1. 관리자 권한으로 **ipaserver**에 로그인합니다.

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

2. **idm\_user\_1**에 대한 정보를 표시합니다.

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

**idm\_user\_1**이라는 사용자는 **IdM**에 있습니다.

#### 4.4. ANSIBLE 플레이북을 사용하여 JSON 파일에서 여러 IDМ 사용자가 있는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM**에 여러 사용자가 있는지 확인하는 방법을 설명합니다. 사용자는 **JSON** 파일에 저장됩니다.

##### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.

##### 절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 **ipaserver**를 이 파일에 정의합니다.

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

- 2.

필요한 작업을 사용하여 **Ansible** 플레이북 파일을 생성합니다. 확인하려는 사용자 데이터를 사용하여 **JSON** 파일을 참조합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/ensure-users-present.ymlfile.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
    - name: Include users.json
      include_vars:
        file: users.json

    - name: Users present
      ipauser:
        ipaadmin_password: MySecret123
        users: "{{ users }}"
```

3.

`users.json` 파일을 생성하고 **IdM** 사용자를 추가합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/user/users.json` 파일에서 예제를 복사하고 수정할 수 있습니다. 예를 들어 사용자 `idm_user_1`, `idm_user_2`, `idm_user_3` 을 만들고 `Password123` 을 `idm_user_1` 의 암호로 추가하려면 다음을 수행합니다.

```
{
  "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.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

## 검증 단계

- **ipa user-show** 명령을 사용하여 사용자 계정이 **IdM**에 있는지 확인할 수 있습니다.

1. 관리자 권한으로 **ipaserver**에 로그인합니다.

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

2. **idm\_user\_1**에 대한 정보를 표시합니다.

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

**idm\_user\_1**이라는 사용자는 **IdM**에 있습니다.

## 4.5. ANSIBLE 플레이북을 사용하여 사용자가 없는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 특정 사용자가 **IdM**에 없는지 확인하는 방법을 설명합니다.

## 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.

## 절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 **ipaserver**를 이 파일에 정의합니다.

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

2.

보장하려는 IdM이 없는 사용자로 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml` 파일에서 예제를 복사하고 수정할 수 있습니다. 예를 들어 `idm_user_1`, `idm_user_2` 및 `idm_user_3` 사용자를 삭제하려면 다음을 수행합니다.

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

  tasks:
  - name: Delete users idm_user_1, idm_user_2, idm_user_3
    ipauser:
      ipaadmin_password: MySecret123
      users:
      - name: idm_user_1
      - name: idm_user_2
      - name: idm_user_3
      state: absent
```

3.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

## 검증 단계

**ipa user-show** 명령을 사용하여 사용자 계정이 IdM에 없는지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver**에 로그인합니다.

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

2.

`idm_user_1`에 대한 정보 요청:

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

`idm_user_1`이라는 사용자는 IdM에 없습니다.

#### 4.6. 추가 리소스

- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-user.md** Markdown 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/user` 디렉터리에서 샘플 **Ansible** 플레이북을 참조하십시오.

## 5장. IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

이 섹션에서는 ID 관리에서 사용자에게 **sudo** 액세스 권한을 부여하는 방법에 대해 설명합니다.

### 5.1. IDM 클라이언트의 SUDO 액세스

시스템 관리자는 루트가 아닌 사용자가 일반적으로 **root** 사용자에게 대해 예약된 관리 명령을 실행할 수 있도록 **sudo** 액세스 권한을 부여할 수 있습니다. 따라서 사용자가 일반적으로 **root** 사용자를 위해 예약한 관리 명령을 수행해야 하는 경우 **sudo** 를 사용하여 명령을 앞에 놓습니다. 암호를 입력한 후 **root** 사용자 인 것처럼 명령이 실행됩니다. 데이터베이스 서비스 계정과 같은 다른 사용자 또는 그룹으로 **sudo** 명령을 실행하려면 **sudo** 규칙에 대해 **RunAs alias** 을 구성할 수 있습니다.

**RHEL(Red Hat Enterprise Linux) 8** 호스트가 **IdM(Identity Management)** 클라이언트로 등록된 경우 다음 방법으로 호스트에서 어떤 **IdM** 사용자를 실행할 수 있는 **sudo** 규칙을 지정할 수 있습니다.

- 로컬로 **/etc/sudoers** 파일
- **IdM** 중앙

이 섹션에서는 **CLI(명령줄 인터페이스)** 및 **IdM 웹 UI**를 사용하여 **IdM** 클라이언트의 중앙 **sudo** 규칙 생성에 대해 설명합니다.

또한 **UNIX** 기반 운영 체제에서 **Kerberos** 서비스에 액세스하고 인증하는 기본 방법인 **GSSAPI(Generic Security Service Application Programming Interface)**를 사용하여 **sudo** 에 대해 암호 없는 인증을 구성할 수도 있습니다. **pam\_sss\_gs.so Pluggable Authentication Module(PAM)**을 사용하여 **SSSD** 서비스를 통해 **GSSAPI** 인증을 호출하여 사용자가 유효한 **Kerberos** 티켓을 사용하여 **sudo** 명령을 인증할 수 있습니다.

추가 리소스

- [sudo 액세스 관리](#)를 참조하십시오.

### 5.2. CLI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

**IdM(Identity Management)**에서는 특정 명령에 대한 **sudo** 액세스 권한을 특정 **IdM** 호스트의 **IdM** 사용자 계정에 부여할 수 있습니다. 먼저 **sudo** 명령을 추가한 다음 하나 이상의 명령에 대한 **sudo** 규칙을 만

됩니다.

예를 들어 `idm_user_reboot sudo` 규칙을 생성하여 `idm_user` 계정에 `idmclient` 시스템에서 `/usr/sbin/reboot` 명령을 실행할 수 있는 권한을 부여하려면 이 절차를 완료합니다.

#### 사전 요구 사항

- **IdM 관리자로 로그인되어 있습니다.**
- 사용자 암호를 생성하여 IdM에서 `idm_user`에 대한 사용자 계정을 생성하고 계정을 잠금 해제했습니다. CLI를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 [명령줄을 사용하여 사용자 추가](#)를 참조하십시오.
- `idmclient` 호스트에 로컬 `idm_user` 계정이 없습니다. `idm_user` 사용자는 로컬 `/etc/passwd` 파일에 나열되어 있지 않습니다.

#### 절차

1. IdM 관리자로 Kerberos 티켓을 검색합니다.

```
[root@idmclient ~]# kinit admin
```

2. `sudo` 명령의 IdM 데이터베이스에 `/usr/sbin/reboot` 명령을 추가합니다.

```
[root@idmclient ~]# ipa sudocmd-add /usr/sbin/reboot
-----
Added Sudo Command "/usr/sbin/reboot"
-----
Sudo Command: /usr/sbin/reboot
```

3. `idm_user_reboot` 라는 `sudo` 규칙을 만듭니다.

```
[root@idmclient ~]# ipa sudorule-add idm_user_reboot
-----
Added Sudo Rule "idm_user_reboot"
-----
Rule name: idm_user_reboot
Enabled: TRUE
```

4.

`/usr/sbin/reboot` 명령을 `idm_user_reboot` 규칙에 추가합니다.

```
[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.

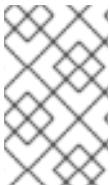
IdM `idmclient` 호스트에 `idm_user_reboot` 규칙을 적용합니다.

```
[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.

`idm_user` 계정을 `idm_user_reboot` 규칙에 추가합니다.

```
[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
-----
```



#### 참고

서버에서 클라이언트로 변경 사항을 전파하는 데 몇 분이 걸릴 수 있습니다.

#### 검증 단계

1.

`idmclient` 호스트에 `idm_user` 계정으로 로그인합니다.

2.

**idm\_user** 계정이 수행할 수 있는 **sudo** 규칙을 표시합니다.

```
[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.

**sudo** 를 사용하여 시스템을 재부팅합니다. 메시지가 표시되면 **idm\_user** 의 암호를 입력합니다.

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

### 5.3. IDM 웹 UI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

**IdM(Identity Management)**에서는 특정 명령에 대한 **sudo** 액세스 권한을 특정 **IdM** 호스트의 **IdM** 사용자 계정에 부여할 수 있습니다. 먼저 **sudo** 명령을 추가한 다음 하나 이상의 명령에 대한 **sudo** 규칙을 만듭니다.

**idm\_user\_reboot sudo** 규칙을 생성하여 **idm\_user** 계정에 **idmclient** 시스템에서 **/usr/sbin/reboot** 명령을 실행할 수 있는 권한을 부여하려면 이 절차를 완료합니다.

#### 사전 요구 사항

- **IdM** 관리자로 로그인되어 있습니다.
- 사용자 암호를 생성하여 **IdM**에서 **idm\_user**에 대한 사용자 계정을 생성하고 계정을 잠금 해제했습니다. 명령줄 인터페이스를 사용하여 새 **IdM** 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄을 사용하여 사용자 추가를 참조하십시오.
-

**idmclient** 호스트에 로컬 **idm\_user** 계정이 없습니다. **idm\_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되어 있지 않습니다.

## 절차

1. **sudo** 명령의 IdM 데이터베이스에 **/usr/sbin/reboot** 명령을 추가합니다.
  - a. 정책 → **Sudo** → **Sudo** 명령으로 이동합니다.
  - b. 오른쪽 상단에서 **Add** 를 클릭하여 **sudo** 명령 추가 대화 상자를 엽니다.
  - c. **sudo:/usr/sbin/reboot** 를 사용하여 사용자가 수행할 수 있는 명령을 입력합니다.

그림 5.1. IdM sudo 명령 추가

The screenshot shows a dialog box titled "Add sudo command" with a close button (X) in the top right corner. Inside the dialog, there are two main input areas: "Sudo Command \*" and "Description". The "Sudo Command" field is a text box containing the text "/usr/sbin/reboot" and is highlighted with a blue border. The "Description" field is a larger text area that is currently empty. Below these fields, there is a note that says "\* Required field". At the bottom of the dialog, there are four buttons: "Add", "Add and Add Another", "Add and Edit", and "Cancel".

- d. 추가를 클릭합니다.
2. **idm\_user** 가 **idmclient** 시스템을 재부팅할 수 있도록 새 **sudo** 명령 항목을 사용하여 **sudo** 규칙을 생성합니다.
  - a. 정책 → **Sudo** → **Sudo** 규칙으로 이동합니다.

- b. 오른쪽 상단에서 **Add** 를 클릭하여 **sudo** 규칙 추가 대화 상자를 엽니다.
- c. **sudo** 규칙의 이름을 입력합니다. **idm\_user\_reboot**.
- d. **추가**를 클릭하고 **편집** 을 클릭합니다.
- e. 사용자를 지정합니다.
  - i. **who** 섹션에서 지정된 사용자 및 그룹 라디오 버튼을 선택합니다.
  - ii. 사용자 카테고리에서 규칙이 하위 섹션에 적용되는 규칙 에서 **Add users into sudo rule "idm\_user\_reboot"** 대화 상자를 엽니다.
  - iii. **Add users into sudo rule "idm\_user\_reboot"** 대화 상자에서 **Available** 열에 **idm\_user** 확인란을 선택하고 **Prospective** 열로 이동합니다.
  - iv. **추가**를 클릭합니다.
- f. 호스트를 지정합니다.
  - i. 액세스 이 호스트 섹션에서 지정된 호스트 및 그룹 라디오 버튼을 선택합니다.
  - ii. 호스트 카테고리에서 이 규칙이 하위 섹션에 적용되면 **Add** 를 클릭하여 **sudo** 규칙 **"idm\_user\_reboot"** 대화 상자에 호스트 추가 대화 상자를 엽니다.
  - iii. 사용 가능 열의 **sudo** 규칙 **"idm\_user\_reboot"** 대화 상자에 호스트 추가 상자에서 **idmclient.idm.example.com** 확인란을 선택하고 **Prospective** 열로 이동합니다.
  - iv. **추가**를 클릭합니다.

g.

명령을 지정합니다.

i.

명령 카테고리에서 규칙은 실행 명령 섹션의 하위 섹션에 적용되며 지정된 명령 및 그룹 라디오 버튼을 선택합니다.

ii.

**Sudo Allow Commands (명령 허용)** 하위 섹션에서 **Add allow sudo commands into sudo rule "idm\_user\_reboot"** 대화 상자가 열립니다.

iii.

**Add allow sudo commands into sudo rule "idm\_user\_reboot"** 대화 상자의 **Available (사용 가능)** 열에서 **/usr/sbin/reboot** 확인란을 선택한 후 **Prospective** 열로 이동합니다.

iv.

**Add** 를 클릭하여 **idm\_sudo\_reboot** 페이지로 돌아갑니다.

## 그림 5.2. IdM sudo 규칙 추가

Who

User category the rule applies to: ☐ Anyone ☒ Specified Users and Groups

|                                              |          |                                                                          |
|----------------------------------------------|----------|--------------------------------------------------------------------------|
| <input type="checkbox"/> Users               | External | <input type="button" value="Delete"/> <input type="button" value="Add"/> |
| <input checked="" type="checkbox"/> idm_user |          |                                                                          |
| <input type="checkbox"/> User Groups         |          | <input type="button" value="Delete"/> <input type="button" value="Add"/> |

Access this host

Host category the rule applies to: ☐ Any Host ☒ Specified Hosts and Groups

|                                                               |          |                                                                          |
|---------------------------------------------------------------|----------|--------------------------------------------------------------------------|
| <input type="checkbox"/> Hosts                                | External | <input type="button" value="Delete"/> <input type="button" value="Add"/> |
| <input checked="" type="checkbox"/> idmclient.idm.example.com |          |                                                                          |
| <input type="checkbox"/> Host Groups                          |          | <input type="button" value="Delete"/> <input type="button" value="Add"/> |

Run Commands

Command category the rule applies to: ☐ Any Command ☒ Specified Commands and Groups

Allow

|                                                                          |                                                                          |
|--------------------------------------------------------------------------|--------------------------------------------------------------------------|
| <input type="checkbox"/> Sudo Allow Commands                             | <input type="button" value="Delete"/> <input type="button" value="Add"/> |
| <input checked="" type="checkbox"/> /usr/sbin/reboot                     |                                                                          |
| <input type="checkbox"/> Sudo Allow Command Groups                       |                                                                          |
| <input type="button" value="Delete"/> <input type="button" value="Add"/> |                                                                          |

h.

왼쪽 상단 모서리에서 저장을 클릭합니다.

새 규칙은 기본적으로 활성화되어 있습니다.



## 참고

서버에서 클라이언트로 변경 사항을 전파하는 데 몇 분이 걸릴 수 있습니다.

1. **idmclient** 에 **idm\_user** 로 로그인합니다.
2. **sudo** 를 사용하여 시스템을 재부팅합니다. 메시지가 표시되면 **idm\_user** 의 암호를 입력합니다.

```
$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

**sudo** 규칙이 올바르게 구성된 경우 시스템이 재부팅됩니다.

#### 5.4. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 CLI에 SUDO 규칙 생성

**IdM**에서는 **RunAs** 별칭을 사용하여 **sudo** 규칙을 구성하여 다른 사용자 또는 그룹으로 **sudo** 명령을 실행할 수 있습니다. 예를 들어 데이터베이스 애플리케이션을 호스팅하는 **IdM** 클라이언트가 있을 수 있으며 해당 애플리케이션에 해당하는 로컬 서비스 계정으로 명령을 실행해야 합니다.

이 예제를 사용하여 **run\_third-party-app\_report** 라는 명령 줄에 **sudo** 규칙을 생성하여 **idmclient** 호스트의 **thirdpartyapp** 서비스 계정으로 **idm\_user** 계정에서 **/opt/ third-party-app/bin/report** 명령을 실행할 수 있습니다.

##### 사전 요구 사항

- **IdM** 관리자로 로그인되어 있습니다.
- 사용자 암호를 생성하여 **IdM**에서 **idm\_user** 에 대한 사용자 계정을 생성하고 계정을 잠금 해제했습니다. **CLI**를 사용하여 새 **IdM** 사용자를 추가하는 방법에 대한 자세한 내용은 [명령줄을 사용하여 사용자 추가](#)를 참조하십시오.
- **idmclient** 호스트에 로컬 **idm\_user** 계정이 없습니다. **idm\_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되어 있지 않습니다.
- **third-party-app** 이라는 사용자 지정 애플리케이션이 **idmclient** 호스트에 설치되어 있습니다.
- **third-party-app** 애플리케이션에 대한 **report** 명령은 **/opt/ third-party-app/bin/report** 디렉터리에 설치됩니다.

- **third-party-app** 애플리케이션에 대한 명령을 실행하기 위해 **thirdpartyapp** 이라는 로컬 서비스 계정을 생성했습니다.

## 절차

1. IdM 관리자로 **Kerberos** 티켓을 검색합니다.

```
[root@idmclient ~]# kinit admin
```

2. **sudo** 명령의 IdM 데이터베이스에 **/opt/third-party-app/bin/report** 명령을 추가합니다.

```
[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. 이름이 **run\_third-party-app\_report** 인 **sudo** 규칙을 만듭니다.

```
[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. **user = <user>** 옵션을 사용하여 **sudorule-add-runasuser** 명령에 대해 **RunAs** 사용자를 지정합니다.

```
[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
-----
```

로컬 서비스 계정 또는 **Active Directory** 사용자와 같은 IdM에 대해 **--groups=\*** 옵션으로 지정된 사용자(또는 그룹)는 외부일 수 있습니다. 그룹 이름에 대해 % 접두사를 추가하지 마십시오.

5.

idm\_user\_reboot 규칙에 /opt/third-party-app/bin/report 명령을 추가합니다.

```
[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.

IdM idmclient 호스트에 run\_third-party-app\_report 규칙을 적용합니다.

```
[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.

idm\_user 계정을 run\_third-party-app\_report 규칙에 추가합니다.

```
[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
```



#### 참고

서버에서 클라이언트로 변경 사항을 전파하는 데 몇 분이 걸릴 수 있습니다.

검증 단계

1. **idmclient** 호스트에 **idm\_user** 계정으로 로그인합니다.
2. 새 **sudo** 규칙을 테스트합니다.
  - a. **idm\_user** 계정이 수행할 수 있는 **sudo** 규칙을 표시합니다.

```
[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. **report** 명령을 **thirdpartyapp** 서비스 계정으로 실행합니다.

```
[idm_user@idmclient ~]$ sudo -u thirdpartyapp /opt/third-party-app/bin/report
[sudo] password for idm_user@idm.example.com:
Executing report...
Report successful.
```

## 5.5. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 IDM WEBUI에 SUDO 규칙 생성

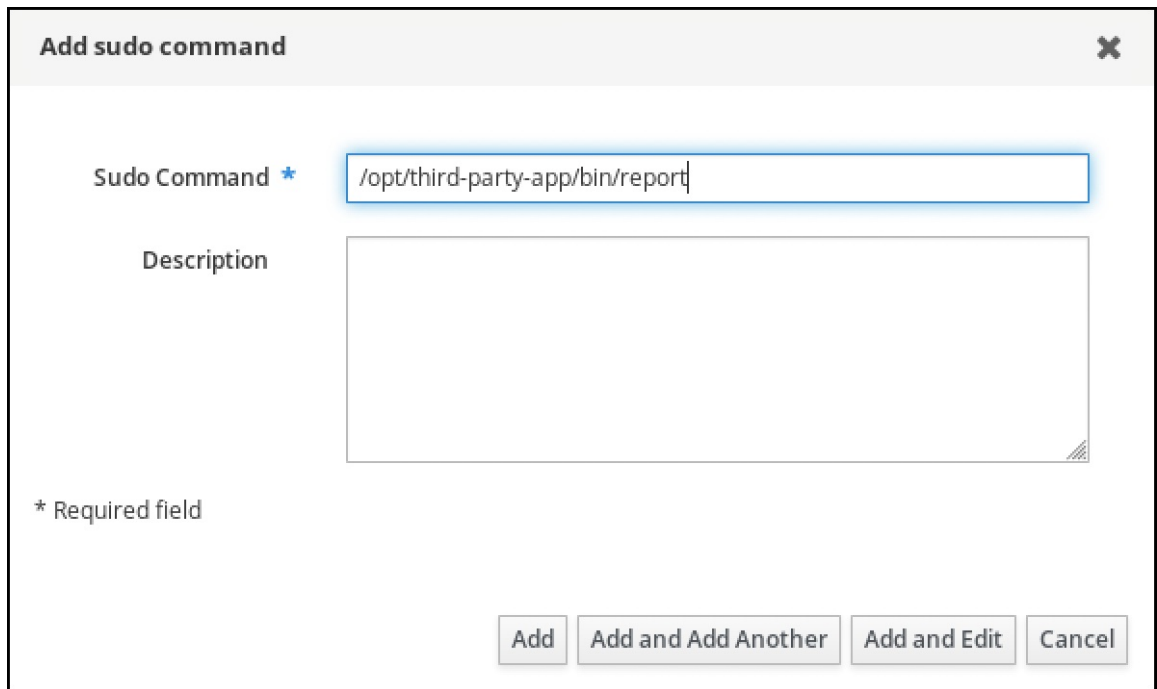
IdM에서는 **RunAs** 별칭을 사용하여 **sudo** 규칙을 구성하여 다른 사용자 또는 그룹으로 **sudo** 명령을 실행할 수 있습니다. 예를 들어 데이터베이스 애플리케이션을 호스팅하는 IdM 클라이언트가 있을 수 있으며 해당 애플리케이션에 해당하는 로컬 서비스 계정으로 명령을 실행해야 합니다.

이 예제를 사용하여 **run\_third-party-app\_report** 라는 IdM WebUI에 **sudo** 규칙을 생성하여 **idmclient** 호스트의 **thirdpartyapp** 서비스 계정으로 **idm\_user** 계정에서 **/opt/third-party-app/bin/report** 명령을 실행할 수 있습니다.

- **IdM 관리자로 로그인되어 있습니다.**
- 사용자 암호를 생성하여 **IdM**에서 **idm\_user**에 대한 사용자 계정을 생성하고 계정을 잠금 해제했습니다. **CLI**를 사용하여 새 **IdM** 사용자를 추가하는 방법에 대한 자세한 내용은 [명령줄을 사용하여 사용자 추가](#)를 참조하십시오.
- **idmclient** 호스트에 로컬 **idm\_user** 계정이 없습니다. **idm\_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되어 있지 않습니다.
- **third-party-app**이라는 사용자 지정 애플리케이션이 **idmclient** 호스트에 설치되어 있습니다.
- **third-party-app** 애플리케이션에 대한 **report** 명령은 **/opt/ third-party-app/bin/report** 디렉터리에 설치됩니다.
- **third-party-app** 애플리케이션에 대한 명령을 실행하기 위해 **thirdpartyapp**이라는 로컬 서비스 계정을 생성했습니다.

## 절차

1. **sudo** 명령의 **IdM** 데이터베이스에 **/opt/third-party-app/bin/report** 명령을 추가합니다.
  - a. 정책 → **Sudo** → **Sudo** 명령으로 이동합니다.
  - b. 오른쪽 상단에서 **Add**를 클릭하여 **sudo** 명령 추가 대화 상자를 엽니다.
  - c. **/opt/third-party-app/bin/report** 명령을 입력합니다.



**Add sudo command** [X]

Sudo Command \*

Description

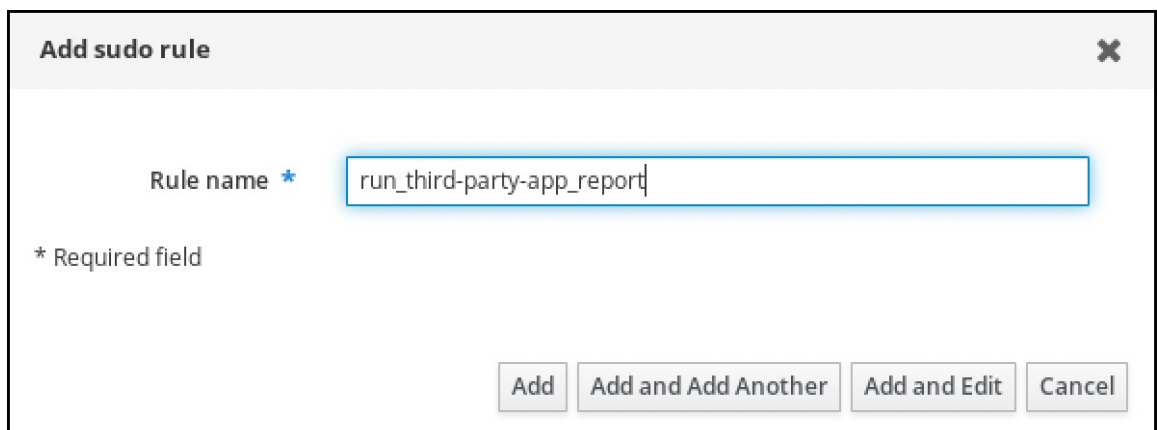
\* Required field

Add Add and Add Another Add and Edit Cancel

- d.   
 추가를 클릭합니다.

2.   
 새 **sudo** 명령 항목을 사용하여 새 **sudo** 규칙을 생성합니다.

- a.   
 정책 → **Sudo** → **Sudo** 규칙으로 이동합니다.
- b.   
 오른쪽 상단에서 **Add** 를 클릭하여 **sudo** 규칙 추가 대화 상자를 엽니다.
- c.   
 **sudo** 규칙의 이름을 입력합니다. **run\_third-party-app\_report**.



**Add sudo rule** [X]

Rule name \*

\* Required field

Add Add and Add Another Add and Edit Cancel

- d.   
 추가를 클릭하고 편집 을 클릭합니다.

e.

사용자를 지정합니다.

i.

**who** 섹션에서 지정된 사용자 및 그룹 라디오 버튼을 선택합니다.

ii.

사용자 카테고리에서 규칙이 하위 섹션에 적용되는 경우 **Add users into sudo rule "run\_third-party-app\_report"** 대화 상자를 클릭합니다.

iii.

**Add users into sudo rule "run\_third-party-app\_report"** 대화 상자에서 **Available** 열에 **idm\_user** 확인란을 선택한 후 **Prospective** 열로 이동합니다.

iv.

추가를 클릭합니다.

f.

호스트를 지정합니다.

i.

**액세스 이 호스트** 섹션에서 지정된 호스트 및 그룹 라디오 버튼을 선택합니다.

ii.

호스트 카테고리에서 이 규칙이 하위 섹션에 적용되면 **Add hosts**를 클릭하여 **sudo** 규칙 **"run\_third-party-app\_report"** 대화 상자에 호스트 추가를 엽니다.

iii.

**Add hosts into sudo rule "run\_third-party-app\_report"** 대화 상자의 **Available** 열에 **idm.idm.example.com** 확인란을 선택하고 **Prospective** 열로 이동합니다.

**Add hosts into sudo rule 'run\_third-party-app\_report'**

Filter available Hosts Filter

| Available                |                         |                                 | Prospective                         |                           |
|--------------------------|-------------------------|---------------------------------|-------------------------------------|---------------------------|
| <input type="checkbox"/> | Hosts                   | <div>&gt;</div> <div>&lt;</div> | <input type="checkbox"/>            | Hosts                     |
| <input type="checkbox"/> | client2.idm.example.com |                                 | <input checked="" type="checkbox"/> | idmclient.idm.example.com |

External

Add Cancel

iv. 추가를 클릭합니다.

g. 명령을 지정합니다.

i. 명령 카테고리에서 규칙은 실행 명령 섹션의 하위 섹션에 적용되며 지정된 명령 및 그룹 라디오 버튼을 선택합니다.

ii. **Sudo Allow Commands (명령 허용)** 하위 섹션에서 **Add allow sudo command into sudo rule "run\_third-party-app\_report"** 대화 상자를 엽니다.

iii. **Add allow sudo commands into sudo rule "run\_third-party-app\_report"** dialog box in the Available column, check the `/opt/third-party-app/bin/report` 체크 박스를 선택하고 **Prospective** 열로 이동합니다.

**Add allow sudo commands into sudo rule 'run\_third-party-app\_report'**

Filter available Sudo Commands Filter

| Available                |               |                                 | Prospective                         |                                 |
|--------------------------|---------------|---------------------------------|-------------------------------------|---------------------------------|
| <input type="checkbox"/> | Sudo Commands | <div>&gt;</div> <div>&lt;</div> | <input type="checkbox"/>            | Sudo Commands                   |
| <input type="checkbox"/> | help          |                                 | <input checked="" type="checkbox"/> | /opt/third-party-app/bin/report |

Add Cancel

iv.

**Add** (추가)를 클릭하여 **run\_third-party-app\_report** 페이지로 돌아갑니다.

h.

**Runas** 사용자를 지정합니다.

i.

**As whom** 섹션에서 지정된 사용자 및 그룹 라디오 버튼을 선택합니다.

ii.

**RunAs Users** 하위 섹션에서 **Add** 를 클릭하여 **Add RunAs users into sudo rule "run\_third-app\_report"** 대화 상자를 엽니다.

iii.

**Add RunAs users into sudo rule "run\_third-party-app\_report"** 대화 상자에서 **External** 상자에 **thirdpartyapp** 서비스 계정을 입력하고 **Prospective** 열로 이동합니다.

iv.

**Add** (추가)를 클릭하여 **run\_third-party-app\_report** 페이지로 돌아갑니다.

i.

왼쪽 상단 모서리에서 저장을 클릭합니다.

새 규칙은 기본적으로 활성화되어 있습니다.

그림 5.3. **sudo** 규칙에 대한 세부 정보

Who

User category the rule applies to: ☐ Anyone ☒ Specified Users and Groups

|                                                   |          |                                              |
|---------------------------------------------------|----------|----------------------------------------------|
| <input type="checkbox"/> Users                    | External | <a href="#">Delete</a> <a href="#">+ Add</a> |
| <input type="checkbox"/> <a href="#">idm_user</a> |          |                                              |

☐ User Groups [Delete](#) [+ Add](#)

Access this host

Host category the rule applies to: ☐ Any Host ☒ Specified Hosts and Groups

|                                                                    |          |                                              |
|--------------------------------------------------------------------|----------|----------------------------------------------|
| <input type="checkbox"/> Hosts                                     | External | <a href="#">Delete</a> <a href="#">+ Add</a> |
| <input type="checkbox"/> <a href="#">idmclient.idm.example.com</a> |          |                                              |

☐ Host Groups [Delete](#) [+ Add](#)

Run Commands

Command category the rule applies to: ☐ Any Command ☒ Specified Commands and Groups

Allow

|                                                                          |                                              |
|--------------------------------------------------------------------------|----------------------------------------------|
| <input type="checkbox"/> Sudo Allow Commands                             | <a href="#">Delete</a> <a href="#">+ Add</a> |
| <input type="checkbox"/> <a href="#">/opt/third-party-app/bin/report</a> |                                              |

☐ Sudo Allow Command Groups [Delete](#) [+ Add](#)

Deny

|                                                   |                                              |
|---------------------------------------------------|----------------------------------------------|
| <input type="checkbox"/> Sudo Deny Commands       | <a href="#">Delete</a> <a href="#">+ Add</a> |
| <input type="checkbox"/> Sudo Deny Command Groups | <a href="#">Delete</a> <a href="#">+ Add</a> |

As Whom

RunAs User category the rule applies to: ☐ Anyone ☒ Specified Users and Groups

|                                                        |          |                                              |
|--------------------------------------------------------|----------|----------------------------------------------|
| <input type="checkbox"/> RunAs Users                   | External | <a href="#">Delete</a> <a href="#">+ Add</a> |
| <input type="checkbox"/> <a href="#">thirdpartyapp</a> | True     |                                              |

☐ Groups of RunAs Users [Delete](#) [+ Add](#)

RunAs Group category the rule applies to: ☐ Any Group ☒ Specified Groups

|                                       |          |                                              |
|---------------------------------------|----------|----------------------------------------------|
| <input type="checkbox"/> RunAs Groups | External | <a href="#">Delete</a> <a href="#">+ Add</a> |
|---------------------------------------|----------|----------------------------------------------|



## 참고

서버에서 클라이언트로 변경 사항을 전파하는 데 몇 분이 걸릴 수 있습니다.

## 검증 단계

1.

**idmclient** 호스트에 **idm\_user** 계정으로 로그인합니다.

2.

새 **sudo** 규칙을 테스트합니다.

a.

**idm\_user** 계정이 수행할 수 있는 **sudo** 규칙을 표시합니다.

```
[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.

**report** 명령을 **thirdpartyapp** 서비스 계정으로 실행합니다.

```
[idm_user@idmclient ~]$ sudo -u thirdpartyapp /opt/third-party-app/bin/report
[sudo] password for idm_user@idm.example.com:
Executing report...
Report successful.
```

## 5.6. IDM 클라이언트에서 SUDO에 대한 GSSAPI 인증 활성화

다음 절차에서는 **pam\_sss\_gss.so** PAM 모듈을 통해 **sudo** 및 **sudo -i** 명령에 대한 **IdM** 클라이언트에 대한 일반 보안 서비스 애플리케이션 프로그램 인터페이스(**GSSAPI**) 인증 활성화에 대해 설명합니다. 이 구성을 통해 **IdM** 사용자는 **Kerberos** 티켓을 사용하여 **sudo** 명령에 인증할 수 있습니다.

### 사전 요구 사항

•

**IdM** 호스트에 적용되는 **IdM** 사용자에게 대한 **sudo** 규칙을 생성했습니다. 이 예제에서는 **idm\_user\_reboot sudo** 규칙을 생성하여 **idm\_user** 계정에 **idmclient** 호스트에서 **/usr/sbin/reboot** 명령을 실행할 수 있는 권한을 부여합니다.

•

**/etc/sss/sss.conf** 파일과 **PAM** 파일을 **/etc/pam.d/** 디렉토리에서 수정하려면 **root** 권한이 필요합니다.

## 절차

1. `/etc/sss/sss.conf` 설정 파일을 엽니다.
2. `[domain/ <domain_name>; ]` 섹션에 다음 항목을 추가합니다.

```
[domain/<domain_name>]
pam_gssapi_services = sudo, sudo-i
```

3. `/etc/sss/sss.conf` 파일을 저장하고 닫습니다.
4. SSSD 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.

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

5. `/etc/pam.d/sudo` PAM 구성 파일을 엽니다.
6. 다음 항목을 `/etc/pam.d/sudo` 파일에서 `auth` 섹션의 첫 번째 행으로 추가합니다.

```
##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. `/etc/pam.d/sudo` 파일을 저장하고 종료합니다.
8. `/etc/pam.d/sudo-i` PAM 구성 파일을 엽니다.
9. 다음 항목을 `/etc/pam.d/sudo-i` 파일에 있는 `auth` 섹션의 첫 번째 행으로 추가합니다.

```
##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.

/etc/pam.d/sudo-i 파일을 저장하고 닫습니다.

## 검증 단계

1.

idm\_user 계정으로 호스트에 로그인합니다.

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

2.

idm\_user 계정으로 티켓이 통합되었는지 확인합니다.

```
[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.

(선택 사항) idm\_user 계정에 대한 Kerberos 자격 증명이 없는 경우 현재 Kerberos 자격 증명을 제거하고 올바른 자격 증명을 요청합니다.

```
[idm_user@idmclient ~]$ kdestroy -A
```

```
[idm_user@idmclient ~]$ kinit idm_user@IDM.EXAMPLE.COM
Password for idm_user@idm.example.com:
```

4.

암호를 지정하지 않고 sudo 를 사용하여 시스템을 재부팅합니다.

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

## 추가 리소스

•

IdM 용어 목록의 GSSAPI 항목

- **IdM 웹 UI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여**
- **CLI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한을 부여합니다.**
- **pam\_sss\_gss(8) 도움말 페이지**
- **sssd.conf (5) 도움말 페이지**

## 5.7. IDM 클라이언트에서 SUDO에 대한 GSSAPI 인증 활성화 및 적용

다음 절차에서는 **pam\_sss\_gss.so** PAM 모듈을 통해 **sudo** 및 **sudo -i** 명령에 대한 IdM 클라이언트에 대한 일반 보안 서비스 애플리케이션 프로그램 인터페이스(**GSSAPI**) 인증 활성화에 대해 설명합니다. 또한 스마트 카드로 로그인한 사용자만 **Kerberos** 티켓을 사용하여 해당 명령에 대해 인증합니다.



### 참고

이 절차를 템플릿으로 사용하여 다른 **PAM** 인식 서비스에 대해 **SSSD**를 사용하여 **GSSAPI** 인증을 구성하고, **Kerberos** 티켓에 연결된 특정 인증 지표가 있는 사용자만 액세스를 제한할 수 있습니다.

### 사전 요구 사항

- IdM 호스트에 적용되는 IdM 사용자에게 대한 **sudo** 규칙을 생성했습니다. 이 예제에서는 **idm\_user\_reboot sudo** 규칙을 생성하여 **idm\_user** 계정에 **idmclient** 호스트에서 **/usr/sbin/reboot** 명령을 실행할 수 있는 권한을 부여합니다.
- **idmclient** 호스트에 대한 스마트 카드 인증을 구성했습니다.
- **/etc/sss/sss.conf** 파일과 **PAM** 파일을 **/etc/pam.d/** 디렉토리에서 수정하려면 **root** 권한이 필요합니다.

### 절차

1. **/etc/sss/sss.conf** 설정 파일을 엽니다.

2.

[domain/ <domain\_name>] 섹션에 다음 항목을 추가합니다.

```
[domain/<domain_name>]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:pkinit, sudo-i:pkinit
```

3.

/etc/sss/sss.conf 파일을 저장하고 닫습니다.

4.

SSSD 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.

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

5.

/etc/pam.d/sudo PAM 구성 파일을 엽니다.

6.

다음 항목을 /etc/pam.d/sudo 파일에서 auth 섹션의 첫 번째 행으로 추가합니다.

```
##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.

/etc/pam.d/sudo 파일을 저장하고 종료합니다.

8.

/etc/pam.d/sudo-i PAM 구성 파일을 엽니다.

9.

다음 항목을 /etc/pam.d/sudo-i 파일에 있는 auth 섹션의 첫 번째 행으로 추가합니다.

```
##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.

**/etc/pam.d/sudo-i** 파일을 저장하고 닫습니다.

## 검증 단계

1.

**idm\_user** 계정으로 호스트에 로그인하고 스마트 카드로 인증합니다.

```
[root@idmclient ~]# ssh -l idm_user@idm.example.com localhost
PIN for smart_card
```

2.

스마트 카드 사용자로 티켓팅 티켓이 있는지 확인합니다.

```
[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.

**idm\_user** 계정이 수행할 수 있는 **sudo** 규칙을 표시합니다.

```
[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.

암호를 지정하지 않고 **sudo** 를 사용하여 시스템을 재부팅합니다.

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

## 추가 리소스

- **PAM 서비스의 GSSAPI 인증을 제어하는 SSSD 옵션**
- **IdM 용어 목록의 GSSAPI 항목**
- **스마트 카드 인증을 위한 ID 관리 구성**
- **Kerberos 인증 표시**
- **IdM 웹 UI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여**
- **CLI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한을 부여합니다.**
- **pam\_sss\_gss(8) 도움말 페이지**
- **sssd.conf (5) 도움말 페이지**

## 5.8. PAM 서비스의 GSSAPI 인증을 제어하는 SSSD 옵션

/etc/sss/sss.conf 구성 파일에 다음 옵션을 사용하여 SSSD 서비스 내에서 GSSAPI 설정을 조정할 수 있습니다.

### pam\_gssapi\_services

SSSD를 사용한 GSSAPI 인증은 기본적으로 비활성화되어 있습니다. 이 옵션을 사용하여 **pam\_sss\_gs.so** PAM 모듈을 사용하여 GSSAPI 인증을 시도할 수 있는 쉘표로 구분된 PAM 서비스 목록을 지정할 수 있습니다. GSSAPI 인증을 명시적으로 비활성화하려면 이 옵션을 - 로 설정합니다.

### pam\_gssapi\_indicators\_map

이 옵션은 IdM(Identity Management) 도메인에만 적용됩니다. 이 옵션을 사용하여 PAM 액세스 권한을 서비스에 부여하는 데 필요한 Kerberos 인증 지표를 나열합니다. 쌍은 < **PAM\_service** > :\_ < **required\_authentication\_indicator** > \_ 형식이어야 합니다.

유효한 인증 지표는 다음과 같습니다.

- 이중 인증을 위한 **OTP**
- **RADIUS** 인증의 경우 **radius**
- **PKINIT**, 스마트 카드 또는 인증서 인증의 **PKINIT**
- 강화된 암호를 위해 강화됨

### pam\_gssapi\_check\_upn

이 옵션은 활성화되며 기본적으로 **true** 로 설정됩니다. 이 옵션을 활성화하면 **SSSD** 서비스에 사용자 이름이 **Kerberos** 자격 증명과 일치해야 합니다. **false** 인 경우 **pam\_sss\_gs.so** PAM 모듈은 필요한 서비스 티켓을 가져올 수 있는 모든 사용자를 인증합니다.

### 예제

다음 옵션을 사용하여 **sudo** 및 **sudo-i** 서비스에 대해 **Kerberos** 인증을 사용하려면 **sudo** 사용자가 일회성 암호로 인증되어야 하며 사용자 이름은 **Kerberos** 주체와 일치해야 합니다. 이러한 설정은 **[pam]** 섹션에 있기 때문에 모든 도메인에 적용됩니다.

```
[pam]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:otp
pam_gssapi_check_upn = true
```

이러한 옵션을 개별 **[domain]** 섹션에서 설정하여 **[pam]** 섹션의 전역 값을 덮어쓸 수도 있습니다. 다음 옵션은 각 도메인에 다른 **GSSAPI** 설정을 적용합니다.

### idm.example.com 도메인의 경우

- **sudo** 및 **sudo-i** 서비스에 대해 **GSSAPI** 인증을 활성화합니다.
- **sudo** 명령에 인증서 또는 스마트 카드 인증 인증 인증 인증 인증 인증이 필요합니다.
- **sudo-i** 명령에 대해 일회성 암호 인증 인증 인증 인증 인증 인증 인증이 필요합니다.

- 일치하는 사용자 이름과 **Kerberos** 주체 적용.

ad.example.com 도메인의 경우

- **sudo** 서비스에 대해서만 **GSSAPI** 인증을 활성화합니다.
- 일치하는 사용자 이름과 보안 주체를 적용하지 마십시오.

```
[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
...
```

추가 리소스

- [Kerberos 인증 표시](#)

## 5.9. SUDO용 GSSAPI 인증 문제 해결

IdM에서 **Kerberos** 티켓을 사용하여 **sudo** 서비스를 인증할 수 없는 경우 다음 시나리오를 사용하여 구성 문제를 해결합니다.

사전 요구 사항

- **sudo** 서비스에 대해 **GSSAPI** 인증을 활성화했습니다. **IdM** 클라이언트에서 **sudo**에 대한 **GSSAPI 인증** 활성화를 참조하십시오.
- **/etc/sss/sss.conf** 파일과 **PAM** 파일을 **/etc/pam.d/** 디렉토리에서 수정하려면 **root** 권한이 필요합니다.

절차

- 다음 오류가 표시되면 **Kerberos** 서비스에서 호스트 이름을 기반으로 서비스 티켓에 대한 유효

바른 영역을 확인할 수 없을 수 있습니다.

### Server not found in Kerberos database

이 경우 `/etc/krb5.conf` Kerberos 설정 파일의 `[domain_realm]` 섹션에 호스트 이름을 직접 추가합니다.

```
[idm-user@idm-client ~]$ cat /etc/krb5.conf
...

[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
server.example.com = EXAMPLE.COM
```

- 다음 오류가 표시되면 Kerberos 자격 증명이 없습니다.

### No Kerberos credentials available

이 경우 `kinit` 유틸리티를 사용하여 Kerberos 자격 증명을 검색하거나 `SSSD`를 사용하여 인증합니다.

```
[idm-user@idm-client ~]$ kinit idm-user@IDM.EXAMPLE.COM
Password for idm-user@idm.example.com:
```

- `/var/log/sss/sssd_pam.log` 로그 파일에 다음 오류가 표시되면 Kerberos 자격 증명이 현재 로그인한 사용자의 사용자 이름과 일치하지 않습니다.

User with UPN [<UPN>] was not found.

UPN [<UPN>] does not match target user [<username>].

이 경우 `SSSD`를 사용하여 인증했는지 확인하거나 `/etc/sss/sssd.conf` 파일에서 `pam_gssapi_check_upn` 옵션 비활성화를 고려하십시오.

```
[idm-user@idm-client ~]$ cat /etc/sss/sssd.conf
...

pam_gssapi_check_upn = false
```

- 추가 문제 해결을 위해 `pam_sss_gss.so` PAM 모듈에 대한 디버깅 출력을 활성화할 수 있습니다.

○

/etc/pam.d/ sudo 및 /etc/pam.d/sudo-i와 같이 PAM 파일의 모든 pam\_sss\_gss.so 항목 끝에 debug 옵션을 추가합니다.

```
[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
```

○

pam\_sss\_gss.so 모듈로 인증을 시도하고 콘솔 출력을 검토합니다. 이 예에서는 사용자에게 Kerberos 자격 증명이 없습니다.

```
[idm-user@idm-client ~]$ sudo ls -l /etc/sss/sss.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
```

#### 5.10. ANSIBLE 플레이북을 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 확인

IdM(Identity Management)에서는 특정 명령에 대한 sudo 액세스 권한이 특정 IdM 호스트의 IdM 사용자 계정에 부여되도록 할 수 있습니다.

**idm\_user\_reboot** 라는 **sudo** 규칙이 있는지 확인하려면 이 절차를 완료합니다. 규칙은 **idm\_user** 에 **idmclient** 시스템에서 **/usr/sbin/reboot** 명령을 실행할 수 있는 권한을 부여합니다.

#### 사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- IdM 관리자 암호를 알고 있습니다.
- 사용자 암호를 생성하여 IdM에서 **idm\_user** 사용자 계정이 있는지 확인하고 계정 잠금을 해제했습니다. 명령줄 인터페이스를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄 을 사용하여 사용자 추가를 참조하십시오.
- 사용자 암호를 생성하여 IdM에서 **idm\_user** 사용자 계정이 있는지 확인하고 계정 잠금을 해제했습니다. 명령줄 인터페이스를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄 을 사용하여 사용자 추가를 참조하십시오.
- 사용자 암호를 생성하여 IdM에서 **idm\_user** 사용자 계정이 있는지 확인하고 계정 잠금을 해제했습니다. 명령줄 인터페이스를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄 을 사용하여 사용자 추가를 참조하십시오.
- **idmclient** 에 로컬 **idm\_user** 계정이 없습니다. **idm\_user** 사용자는 **idmclient** 의 **/etc/passwd** 파일에 나열되지 않습니다.

#### 절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 **ipaservers** 를 정의합니다.

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

2. 하나 이상의 **sudo** 명령을 추가합니다.
  - a. **sudo** 명령의 IdM 데이터베이스에 **/usr/sbin/reboot** 명령이 있는지 확인하는 **ensure-reboot-sudocmd-is-present.yml** Ansible 플레이북을 생성합니다. 이 단계를 단순화하기

위해 `/usr/share/doc/ansible-freeipa/playbooks/sudocmd/ensure-sudocmd-is-present.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Playbook to manage sudo command
  hosts: ipaserver
  become: true

  tasks:
    # Ensure sudo command is present
    - ipasudocmd:
        ipadmin_password: MySecret123
        name: /usr/sbin/reboot
        state: present
```

b.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-reboot-sudocmd-is-present.yml
```

3.

명령을 참조하는 **sudo** 규칙을 생성합니다.

a.

**sudo** 명령 항목을 사용하여 **sudo** 규칙이 있는지 확인하는 **ensure-sudorule-for-idmuser-idmclient-is-present.yml** Ansible 플레이북을 만듭니다. **sudo** 규칙을 사용하면 **idm\_user**가 **idmclient** 시스템을 재부팅할 수 있습니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/sudorule/ensure-sudorule-is-present.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Tests
  hosts: ipaserver
  become: true

  tasks:
    # Ensure a sudorule is present granting idm_user the permission to run
    /usr/sbin/reboot on idmclient
    - ipasudorule:
        ipadmin_password: MySecret123
        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.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-sudorule-for-idmuser-on-idmclient-is-
present.yml
```

## 검증 단계

**idm\_user** 가 **sudo** 를 사용하여 **idmclient** 를 재부팅할 수 있는지 확인하여 **IdM** 서버에 있는지 확인하는 **sudo** 규칙을 **idmclient** 에서 테스트하십시오. 서버에서 변경한 내용이 클라이언트에 적용되는 데 몇 분 정도 걸릴 수 있습니다.

1.

**idmclient** 에 **idm\_user** 로 로그인합니다.

2.

**sudo** 를 사용하여 시스템을 재부팅합니다. 메시지가 표시되면 **idm\_user** 의 암호를 입력합니다.

```
$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

**sudo** 가 올바르게 구성된 경우 시스템이 재부팅됩니다.

## 추가 리소스

- 

**/usr/share/doc/ansible-freeipa/** 디렉터리에서 **README-sudocmdgroup.md**, **README-sudorule.md** 파일을 참조하십시오.

## 6장. LDAPMODIFY를 사용하여 외부 IDM 사용자 관리

**ldapmodify** 및 **ldapdelete** 유틸리티를 사용하여 **CLI**(명령줄 인터페이스)에서 **ID 관리(IdM) LDAP**를 직접 수정할 수 있습니다. 유틸리티에서는 디렉터리 콘텐츠 추가, 편집 및 삭제에 필요한 모든 기능을 제공합니다. 이러한 유틸리티를 사용하여 서버의 구성 항목과 사용자 항목의 데이터를 모두 관리할 수 있습니다. 유틸리티를 사용하여 하나 이상의 디렉터리의 대규모 관리를 수행하는 스크립트를 작성할 수도 있습니다.

### 6.1. IDM 사용자 계정을 외부에서 관리하기 위한 템플릿

이 섹션에서는 **IdM**의 다양한 사용자 관리 작업에 대한 템플릿을 설명합니다. 템플릿은 다음 목표를 달성하기 위해 **ldapmodify**를 사용하여 수정해야 하는 속성을 표시합니다.

- 새 스테이지 사용자 추가
- 사용자 특성 수정
- 사용자 활성화
- 사용자 비활성화
- 사용자 보존

템플릿은 **LDAP Data Interchange Format(LDIF)**으로 포맷됩니다. **LDIF**는 **LDAP** 디렉토리 콘텐츠 및 업데이트 요청을 나타내는 표준 텍스트 데이터 교환 형식입니다.

템플릿을 사용하여 **IdM** 사용자 계정을 관리하도록 프로비저닝 시스템의 **LDAP** 공급자를 구성할 수 있습니다.

자세한 예제 절차는 다음 섹션을 참조하십시오.

- **LDIF** 파일에 정의된 **IdM** 단계 사용자 추가

- **Idapmodify를 사용하여 CLI에서 직접 IdM 스테이지 사용자 추가**
- **Idapmodify를 사용하여 IdM 사용자 보존**

새 스테이지 사용자를 추가하기 위한 템플릿

- **UID 및 GID가 자동으로 할당된 사용자를 추가하는 템플릿입니다. 생성된 항목의 DN(고유 이름)은 uid=user\_login 으로 시작해야 합니다.**

```
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
```

- **UID 및 GID가 할당된 사용자를 추가하기 위한 템플릿:**

```
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
```

스테이징 사용자를 추가할 때 **IdM** 오브젝트 클래스를 지정할 필요는 없습니다. **IdM**은 사용자가 활성화된 후 이러한 클래스를 자동으로 추가합니다.

기존 사용자를 수정하기 위한 템플릿

- **사용자 특성 수정:**

```
dn: distinguished_name
changetype: modify
replace: attribute_to_modify
attribute_to_modify: new_value
```

•

사용자 비활성화:

```
dn: distinguished_name
changetype: modify
replace: nsAccountLock
nsAccountLock: TRUE
```

•

사용자 활성화:

```
dn: distinguished_name
changetype: modify
replace: nsAccountLock
nsAccountLock: FALSE
```

**nssAccountLock** 속성을 업데이트해도 준비 및 보존된 사용자에게는 영향을 미치지 않습니다. 업데이트 작업이 성공적으로 완료되었지만 특성 값은 **nssAccountLock: TRUE**로 유지됩니다.

•

사용자 보존:

```
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
```

## 참고

사용자를 수정하기 전에 사용자 로그인을 검색하여 사용자의 고유 이름(DN)을 가져옵니다. 다음 예에서 **`user_allowed_to_modify_user_entries`** 사용자는 사용자 및 그룹 정보를 수정할 수 있는 사용자입니다(예: **`activator`** 또는 **`IdM`** 관리자). 이 예제의 암호는 이 사용자 암호입니다.

```
[...]
# 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
```

## 6.2. IdM 그룹 계정을 외부에서 관리하기 위한 템플릿

이 섹션에서는 **`IdM`**의 다양한 사용자 그룹 관리 작업에 대한 템플릿을 설명합니다. 템플릿은 다음 목표를 달성하기 위해 **`ldapmodify`**를 사용하여 수정해야 하는 속성을 표시합니다.

- 새 그룹 만들기
- 기존 그룹 삭제
- 그룹에 멤버 추가
- 그룹에서 멤버 제거

템플릿은 **`LDAP Data Interchange Format(LDIF)`**으로 포맷됩니다. **`LDIF`**는 **`LDAP`** 디렉토리 콘텐츠 및 업데이트 요청을 나타내는 표준 텍스트 데이터 교환 형식입니다.

템플릿을 사용하여 **`IdM`** 그룹 계정을 관리하도록 프로비저닝 시스템의 **`LDAP`** 공급자를 구성할 수 있습니다.

### 새 그룹 만들기

```
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
```

### 그룹 수정

- 기존 그룹 삭제:

```
dn: group_distinguished_name
changetype: delete
```

- 그룹에 멤버를 추가:

```
dn: group_distinguished_name
changetype: modify
add: member
member: uid=user_login,cn=users,cn=accounts,dc=idm,dc=example,dc=com
```

스테이징 또는 보존된 사용자를 그룹에 추가하지 마십시오. 업데이트 작업이 성공적으로 완료되었지만 사용자는 그룹의 멤버로 업데이트되지 않습니다. 활성 사용자만 그룹에 속할 수 있습니다.

- 그룹에서 멤버를 제거:

```
dn: distinguished_name
changetype: modify
delete: member
member: uid=user_login,cn=users,cn=accounts,dc=idm,dc=example,dc=com
```

## 참고

그룹을 수정하기 전에 그룹의 이름으로 검색하여 그룹의 고유 이름(DN)을 가져옵니다.

```
# ldapsearch -Y GSSAPI -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
```

### 6.3. LDAPMODIFY를 사용하여 IDM 사용자 보존

이 섹션에서는 **ldapmodify** 을 사용하여 **IdM** 사용자를 유지하는 방법에 대해 설명합니다. 즉, 직원이 퇴사한 후 사용자 계정을 비활성화하는 방법.

#### 사전 요구 사항

- 사용자를 유지하기 위해 역할이 있는 **IdM** 사용자로 인증할 수 있습니다.

#### 절차

1. 사용자가 보존하려면 역할이 있는 **IdM** 사용자로 로그인합니다.

```
$ kinit admin
```

2. **ldapmodify** 명령을 입력하고 인증에 사용할 **SASL(Simple Authentication and Security Layer)** 메커니즘으로 **GSSAPI(Generic Security Services API)**를 지정합니다.

```
# ldapmodify -Y GSSAPI
SASL/GSSAPI authentication started
SASL username: admin@IDM.EXAMPLE.COM
SASL SSF: 256
SASL data security layer installed.
```

3. 보존하려는 사용자의 **dn** 을 입력합니다.

```
dn: uid=user1,cn=users,cn=accounts,dc=idm,dc=example,dc=com
```

4. 수행하려는 변경 유형으로 **modrdn** 을 입력합니다.

```
changetype: modrdn
```

5. 사용자의 **newrdn** 을 지정합니다.

```
newrdn: uid=user1
```

6. 사용자를 보존할 것임을 나타냅니다.

```
deleteoldrdn: 0
```

7. 새로운 우수한 **DN** 을 지정합니다.

```
newsuperior: cn=deleted
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
```

사용자를 보존하면 디렉터리 정보 트리(DIT)의 항목을 새 위치로 이동합니다. 따라서 새 상위 항목의 **DN**을 새로운 우수한 **DN**으로 지정해야 합니다.

8. **Enter** 를 다시 눌러 항목의 끝인지 확인합니다.

```
[Enter]
```

```
modifying rdn of entry
"uid=user1,cn=users,cn=accounts,dc=idm,dc=example,dc=com"
```

9. **Ctrl + C** 를 사용하여 연결을 종료합니다.

#### 검증 단계

- 보존된 모든 사용자를 나열하여 사용자가 보존되었는지 확인합니다.

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

## 7장. 사용자의 외부 프로비저닝을 위해 IDM 구성

시스템 관리자는 ID 관리를 위한 외부 솔루션으로 사용자 프로비저닝을 지원하도록 **IdM(Identity Management)**을 구성할 수 있습니다.

**ipa** 유틸리티를 사용하는 대신 외부 프로비저닝 시스템의 관리자는 **ldapmodify** 유틸리티를 사용하여 **IdM LDAP**에 액세스할 수 있습니다. 관리자는 **ldapmodify**를 사용하거나 **LDIF** 파일을 사용하여 **CLI**에서 개별 스테이지 사용자를 추가할 수 있습니다.

**IdM** 관리자는 검증된 사용자만 추가하도록 외부 프로비저닝 시스템을 완전히 신뢰할 수 있다는 가정입니다. 그러나 동시에 외부 프로비저닝 시스템의 관리자에게 새 활성 사용자를 직접 추가할 수 있도록 **User Administrator**의 **IdM** 역할을 할당하지 않아야 합니다.

외부 프로비저닝 시스템에서 생성한 스테이징 사용자를 활성 사용자로 자동으로 이동하도록 **스크립트**를 구성할 수 있습니다.

이 섹션에서는 다음 섹션이 포함되어 있습니다.

1. 외부 프로비저닝 시스템을 사용하도록 **IdM(Identity Management)**을 준비하여 **IdM**에 스테이징 사용자를 추가합니다.
2. 외부 프로비저닝 시스템에서 추가한 사용자를 스테이지에서 활성 사용자로 이동하는 **스크립트**를 생성합니다.
3. 외부 프로비저닝 시스템을 사용하여 **IdM** 스테이징 사용자를 추가합니다. 두 가지 방법으로 이 작업을 수행할 수 있습니다.
  - **LDIF** 파일을 사용하여 **IdM** 스테이징 사용자 추가
  - **ldapmodify**를 사용하여 **CLI**에서 직접 **IdM** 스테이징 사용자를 추가합니다.

### 7.1. 스테이징 사용자 계정의 자동 활성화를 위해 IDM 계정 준비

다음 절차에서는 외부 프로비저닝 시스템에서 사용할 두 개의 **IdM** 사용자 계정을 구성하는 방법을 보여줍니다. 적절한 암호 정책이 있는 그룹에 계정을 추가하여 외부 프로비저닝 시스템을 활성화하여 **IdM**의

사용자 프로비저닝을 관리할 수 있습니다. 다음에서는 외부 시스템에서 스테이징 사용자를 추가하는 데 사용할 사용자 계정의 이름은 **provisionator** 입니다. 스테이지 사용자를 자동으로 활성화하는 데 사용할 사용자 계정은 활성화 기라고 합니다.

#### 사전 요구 사항

- 절차를 수행하는 호스트는 **IdM**에 등록됩니다.

#### 절차

1. **IdM** 관리자로 로그인합니다.

```
$ kinit admin
```

2. 스테이징 사용자를 추가할 권한이 있는 **provisionator** 라는 사용자를 생성합니다.

- a. **provisionator** 사용자 계정을 추가합니다.

```
$ ipa user-add provisionator --first=provisioning --last=account --password
```

- a. 프로비저닝기 사용자에게 필요한 권한을 부여합니다.

- i. 스테이징 사용자 추가를 관리하기 위해 사용자 지정 역할인 시스템 프로비저닝 을 생성합니다.

```
$ ipa role-add --desc "Responsible for provisioning stage users" "System Provisioning"
```

- ii. 스테이지 사용자 프로비저닝 권한을 역할에 추가합니다. 이 권한은 스테이징 사용자를 추가할 수 있는 기능을 제공합니다.

```
$ ipa role-add-privilege "System Provisioning" --privileges="Stage User Provisioning"
```

- iii. 역할에 **provisionator** 사용자를 추가합니다.

```
$ ipa role-add-member --users=provisionator "System Provisioning"
```

iv.

IdM에 프로젝터가 있는지 확인합니다.

```
$ 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.

사용자 계정을 관리할 수 있는 권한을 가진 사용자 **activator** 를 만듭니다.

a.

활성화기 사용자 계정을 추가합니다.

```
$ ipa user-add activator --first=activation --last=account --password
```

b.

기본 사용자 관리자 역할에 사용자를 추가하여 **activator** 사용자에게 필요한 권한을 부여합니다.

```
$ ipa role-add-member --users=activator "User Administrator"
```

4.

애플리케이션 계정에 대한 사용자 그룹을 생성합니다.

```
$ ipa group-add application-accounts
```

5.

그룹의 암호 정책을 업데이트합니다. 다음 정책은 계정의 암호 만료 및 잠금을 방지하지만 복잡한 암호를 요구하는 경우 발생할 수 있는 위험을 보완합니다.

```
$ ipa pwpolicy-add application-accounts --maxlife=10000 --minlife=0 --history=0 --minclasses=4 --minlength=8 --priority=1 --maxfail=0 --failinterval=1 --lockouttime=0
```

6.

(선택 사항) IdM에 암호 정책이 있는지 확인합니다.

```
$ ipa pwpolicy-show application-accounts
Group: application-accounts
Max lifetime (days): 10000
Min lifetime (hours): 0
History size: 0
[...]
```

7.

애플리케이션 계정의 그룹에 프로비저닝 및 활성화 계정을 추가합니다.

```
$ ipa group-add-member application-accounts --users={provisionator,activator}
```

8.

사용자 계정의 암호를 변경합니다.

```
$ kpasswd provisionator
$ kpasswd activator
```

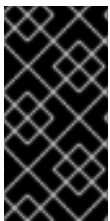
새 IdM 사용자 암호가 즉시 만료되므로 암호를 변경해야 합니다.

추가 리소스:

- [명령줄을 사용하여 사용자 계정](#) 관리를 참조하십시오.
- [사용자 별 권한 정의](#)를 참조하십시오.
- [IdM 암호 정책 정의](#)를 참조하십시오.

## 7.2. IDM 단계 사용자 계정의 자동 활성화 구성

다음 절차에서는 스테이지 사용자를 활성화하는 스크립트를 만드는 방법을 보여줍니다. 시스템은 지정된 시간 간격으로 자동으로 스크립트를 실행합니다. 이렇게 하면 새 사용자 계정이 자동으로 활성화되고 생성된 직후 사용할 수 있습니다.



### 중요

이 절차에서는 외부 프로비저닝 시스템의 소유자가 이미 사용자를 검증했으며 스크립트가 IdM에 추가되기 전에 IdM 측에 추가 검증이 필요하지 않다고 가정합니다.

IdM 서버 중 하나에서 활성화 프로세스를 활성화하는 것으로 충분합니다.

사전 요구 사항

- IdM에 프로비저닝 및 활성화 기 계정이 있습니다. 자세한 내용은 [스테이지 사용자 계정의 자](#)

동 활성화를 위한 **IdM** 계정 준비를 참조하십시오.

- 프로시저를 실행 중인 **IdM** 서버에 대한 **root** 권한이 있어야 합니다.
- **IdM** 관리자로 로그인되어 있습니다.
- 외부 프로비저닝 시스템을 신뢰합니다.

## 절차

1. 활성화 계정에 대한 키탭 파일을 생성합니다.

```
# ipa-getkeytab -s server.idm.example.com -p "activator" -k /etc/krb5.ipa-activation.keytab
```

둘 이상의 **IdM** 서버에서 활성화 프로세스를 활성화하려면 하나의 서버에만 키탭 파일을 생성합니다. 그런 다음 키탭 파일을 다른 서버에 복사합니다.

2. 모든 사용자를 활성화할 다음 콘텐츠를 사용하여 **/usr/local/sbin/ipa-activate-all** 스크립트를 생성합니다.

```
#!/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. **ipa-activate-all** 스크립트의 권한 및 소유권을 편집하여 실행 가능하게 만듭니다.

```
# chmod 755 /usr/local/sbin/ipa-activate-all
# chown root:root /usr/local/sbin/ipa-activate-all
```

4. 다음 콘텐츠를 사용하여 **systemd** 장치 파일 **/etc/systemd/system/ipa-activate-all.service**를 만듭니다.

```
[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. 다음 콘텐츠를 사용하여 **systemd** 타이머 **/etc/systemd/system/ipa-activate-all.timer** 를 만듭니다.

```
[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. 새 구성을 다시 로드합니다.

```
# systemctl daemon-reload
```

7. **ipa-activate-all.timer** 를 활성화합니다.

```
# systemctl enable ipa-activate-all.timer
```

8. **ipa-activate-all.timer** 를 시작합니다.

```
# systemctl start ipa-activate-all.timer
```

9. (선택 사항) **ipa-activate-all.timer** 데몬이 실행 중인지 확인합니다.

```
# 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.
```

### 7.3. LDIF 파일에 정의된 IDM 단계 사용자 추가

이 섹션에서는 외부 프로비저닝 시스템의 관리자가 **IdM LDAP**에 액세스하고 **LDIF** 파일을 사용하여 스테이지 사용자를 추가하는 방법을 설명합니다. 아래 예제에서는 하나의 단일 사용자를 추가하는 방법을 보여주지만 여러 사용자를 대량 모드로 하나의 파일에 추가할 수 있습니다.

#### 사전 요구 사항

- **IdM** 관리자가 프로젝터 계정 과 암호를 생성했습니다. 자세한 내용은 [스테이지 사용자 계정의 자동 활성화를 위한 IdM 계정 준비](#)를 참조하십시오.
- 외부 관리자는 프로젝터 계정의 암호를 알고 있습니다.
- **LDAP** 서버에서 **IdM** 서버에 **SSH**를 수행할 수 있습니다.
- **IdM** 단계 사용자가 사용자 라이프사이클의 올바른 처리를 허용해야 하는 최소한의 속성 세트를 제공할 수 있습니다.
  - 고유 이름 (**dn**)
  - 일반 이름 (**cn**)
  - 마지막 이름 (**sn**)
  - 이메일 주소

#### 절차

1. 외부 서버에서 새 사용자에 대한 정보가 포함된 **LDIF** 파일을 만듭니다.

```
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.

LDIF 파일을 외부 서버에서 IdM 서버로 전송합니다.

```
$ scp add-stageidmuser.ldif provisionator@server.idm.example.com:/provisionator/
Password:
add-stageidmuser.ldif                                100% 364
217.6KB/s 00:00
```

3.

SSH 프로토콜을 사용하여 IdM 서버에 프로비저닝 기로 연결합니다.

```
$ ssh provisionator@server.idm.example.com
Password:
[provisionator@server ~]$
```

4.

IdM 서버에서 프로비저닝자 계정에 대한 Kerberos 티켓(TGT)을 받으십시오.

```
[provisionator@server ~]$ kinit provisionator
```

5.

Idapadd 명령과 -f 옵션 및 LDIF 파일의 이름을 입력합니다. IdM 서버의 이름과 포트 번호를 지정합니다.

```
~]$ Idapadd -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"
```

#### 7.4. LDAPMODIFY를 사용하여 CLI에서 직접 IDM 스테이지 사용자 추가

이 섹션에서는 외부 프로비저닝 시스템의 관리자가 IdM(Identity Management) LDAP에 액세스하고 Idapmodify 유틸리티를 사용하여 스테이지 사용자를 추가하는 방법을 설명합니다.

##### 사전 요구 사항

•

IdM 관리자가 프로젝터 계정과 암호를 생성했습니다. 자세한 내용은 [스테이지 사용자 계정의 자동 활성화를 위한 IdM 계정 준비](#)를 참조하십시오.

- 외부 관리자는 프로젝트 계정의 암호를 알고 있습니다.
- LDAP 서버에서 IdM 서버에 SSH를 수행할 수 있습니다.
- IdM 단계 사용자가 사용자 라이프사이클의 올바른 처리를 허용해야 하는 최소한의 속성 세트를 제공할 수 있습니다.
  - 고유 이름 (dn)
  - 일반 이름 (cn)
  - 마지막 이름 (sn)
  - 이메일 주소

#### 절차

1. IdM ID 및 인증 정보를 사용하여 IdM 서버에 연결하려면 SSH 프로토콜을 사용합니다.

```
$ ssh provisionator@server.idm.example.com
Password:
[provisionator@server ~]$
```

2. 프로비저닝기 계정의 TGT, 새 단계 사용자를 추가할 역할이 있는 IdM 사용자를 가져옵니다.

```
$ kinit provisionator
```

3. Idapmodify 명령을 입력하고 인증에 사용할 SASL(Simple Authentication and Security Layer) 메커니즘으로 GSSAPI(Generic Security Services API)를 지정합니다. IdM 서버 이름과 포트를 지정합니다.

```
# Idapmodify -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.

추가할 사용자의 **dn** 을 입력합니다.**Enter the dn of the user you are adding:**

**dn: uid=stageuser,cn=staged**  
**users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com**

5.

수행하는 변경 사항의 유형으로 **add** 를 입력합니다.

**changetype: add**

6.

사용자 라이프 사이클의 올바른 처리를 허용하는 데 필요한 **LDAP** 오브젝트 클래스 카테고리를 지정합니다.

**objectClass: top**  
**objectClass: inetorgperson**

추가 오브젝트 클래스를 지정할 수 있습니다.

7.

사용자의 **uid** 를 입력합니다.

**uid: stageuser**

8.

사용자의 **cn** 을 입력합니다.

**cn: Babs Jensen**

9.

사용자 이름을 입력합니다.

**sn: Jensen**

10.

**Enter** 를 다시 눌러 항목의 끝인지 확인합니다.

**[Enter]**

**adding new entry "uid=stageuser,cn=staged**

```
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com"
```

11.

**Ctrl + C** 를 사용하여 연결을 종료합니다.

## 검증 단계

스테이징 항목의 콘텐츠를 확인하여 프로비저닝 시스템이 필요한 모든 **POSIX** 속성을 추가하고 스테이징 항목을 활성화할 준비가 되었는지 확인합니다.

- 새 스테이지 사용자의 **LDAP** 속성을 표시하려면 **ipa stageuser-show --all --raw** 명령을 입력합니다.

```
$ 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.

사용자는 **nsaccountlock** 속성에서 명시적으로 비활성화 됩니다.

## 7.5. 추가 리소스

- 외부에서 **IdM** 사용자를 관리하기 위해 **ldapmodify**를 사용하여 를 참조하십시오.

## 8장. CLI를 사용하여 IDM에서 셀프 서비스 규칙 관리

이 장에서는 **IdM(Identity Management)**에 셀프 서비스 규칙을 소개하고 **CLI(명령줄 인터페이스)**에서 셀프 서비스 액세스 규칙을 생성하고 편집하는 방법을 설명합니다.

### 8.1. IDM의 셀프 서비스 액세스 제어

셀프 서비스 액세스 제어 규칙은 **IdM** 디렉터리 서버 항목에서 수행할 수 있는 **IdM(Identity Management)** 엔티티의 작업을 정의합니다. 예를 들어 **IdM** 사용자는 자체 암호를 업데이트할 수 있습니다.

이 제어 방법을 사용하면 인증된 **IdM** 엔티티가 **LDAP** 항목 내에서 특정 속성을 편집할 수 있지만 전체 항목에 대한 작업 추가 또는 삭제는 허용하지 않습니다.



#### 주의

셀프 서비스 액세스 제어 규칙으로 작업할 때는 주의해야 합니다. 액세스 제어 규칙을 부적절하게 구성하면 엔티티의 권한이 실수로 상승할 수 있습니다.

### 8.2. CLI를 사용하여 셀프 서비스 규칙 생성

이 절차에서는 **CLI(명령줄 인터페이스)**를 사용하여 **IdM**에서 셀프 서비스 액세스 규칙 생성에 대해 설명합니다.

#### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조](#) 하십시오.

#### 절차

-

셀프 서비스 규칙을 추가하려면 **ipa selfservice-add** 명령을 사용하고 다음 두 가지 옵션을 지정합니다.

**--permissions**

**ACI(Access Control Instruction)**에서 부여하는 읽기 및 쓰기 권한을 설정합니다.

**--attrs**

이 **ACI**가 권한을 부여하는 전체 속성 목록을 설정합니다.

예를 들어 사용자가 자체 이름 세부 정보를 수정할 수 있는 셀프 서비스 규칙을 생성하려면 다음을 수행합니다.

```
$ 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
```

### 8.3. CLI를 사용하여 셀프 서비스 규칙 편집

이 절차에서는 **CLI(명령줄 인터페이스)**를 사용하여 **IdM**에서 셀프 서비스 액세스 규칙 편집을 설명합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조](#) 하십시오.

절차

1. **선택 사항:** **ipa selfservice-find** 명령을 사용하여 기존 셀프 서비스 규칙을 표시합니다.
2. **선택 사항:** **ipa selfservice-show** 명령을 사용하여 수정하려는 셀프 서비스 규칙에 대한 세부

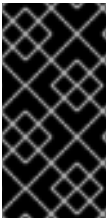
정보를 표시합니다.

3.

**ipa selfservice-mod** 명령을 사용하여 셀프 서비스 규칙을 편집합니다.

예를 들면 다음과 같습니다.

```
$ 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
```



중요

**ipa selfservice-mod** 명령을 사용하면 이전에 정의한 권한 및 속성을 덮어쓰므로, 정의하려는 새 권한 및 속성과 함께 기존 권한 및 속성 목록이 항상 포함됩니다.

검증 단계

•

**ipa selfservice-show** 명령을 사용하여 편집한 셀프 서비스 규칙을 표시합니다.

```
$ 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
```

#### 8.4. CLI를 사용하여 셀프 서비스 규칙 삭제

이 절차에서는 **CLI**(명령줄 인터페이스)를 사용하여 **IdM**에서 셀프 서비스 액세스 규칙 삭제를 설명합니다.

사전 요구 사항

•

**IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

- 활성 **Kerberos** 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조](#) 하십시오.

#### 절차

- **ipa selfservice-del** 명령을 사용하여 셀프 서비스 규칙을 삭제합니다.

예를 들면 다음과 같습니다.

```
$ ipa selfservice-del "Users can manage their own name details"
```

```
Deleted selfservice "Users can manage their own name details"
```

#### 검증 단계

- **ipa selfservice-find** 명령을 사용하여 모든 셀프 서비스 규칙을 표시합니다. 방금 삭제한 규칙이 누락되어야 합니다.

## 9장. IDM 웹 UI를 사용하여 셀프 서비스 규칙 관리

이 장에서는 **IdM(Identity Management)**에 셀프 서비스 규칙을 소개하고 웹 인터페이스(**IdM Web UI**)에서 셀프 서비스 액세스 규칙을 만들고 편집하는 방법을 설명합니다.

### 9.1. IDM의 셀프 서비스 액세스 제어

셀프 서비스 액세스 제어 규칙은 **IdM** 디렉터리 서버 항목에서 수행할 수 있는 **IdM(Identity Management)** 엔티티의 작업을 정의합니다. 예를 들어 **IdM** 사용자는 자체 암호를 업데이트할 수 있습니다.

이 제어 방법을 사용하면 인증된 **IdM** 엔티티가 **LDAP** 항목 내에서 특정 속성을 편집할 수 있지만 전체 항목에 대한 작업 추가 또는 삭제는 허용하지 않습니다.



#### 주의

셀프 서비스 액세스 제어 규칙으로 작업할 때는 주의해야 합니다. 액세스 제어 규칙을 부적절하게 구성하면 엔티티의 권한이 실수로 상승할 수 있습니다.

### 9.2. IDM 웹 UI를 사용하여 셀프 서비스 규칙 생성

이 절차에서는 웹 인터페이스(**IdM Web UI**)를 사용하여 **IdM**에서 셀프 서비스 액세스 규칙을 생성하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **IdM 웹 UI**에 로그인되어 있습니다. 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.

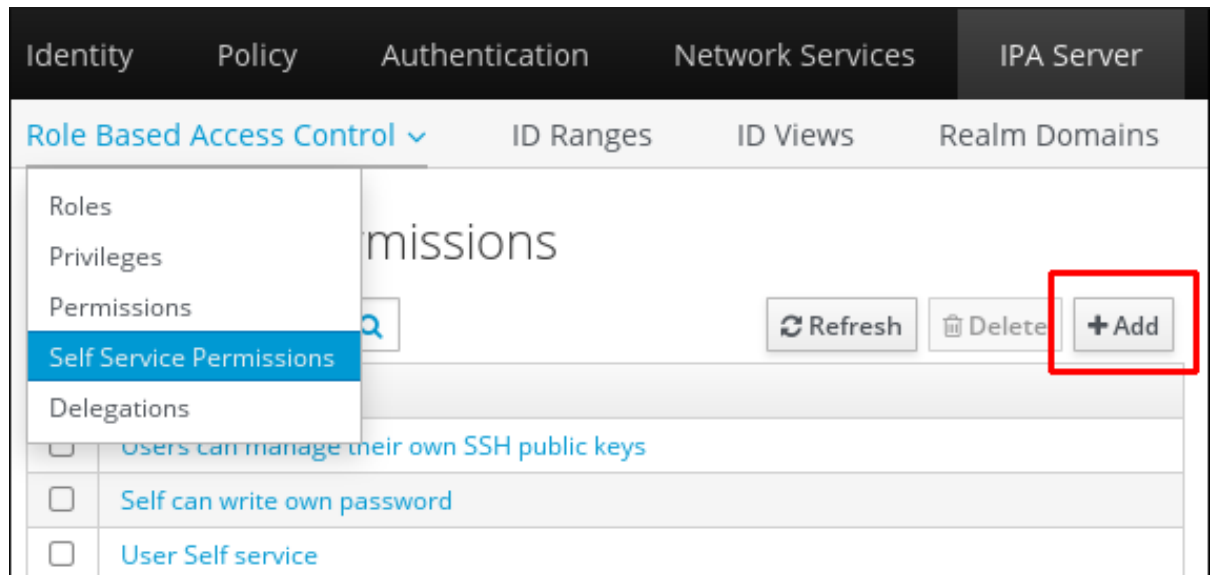
#### 절차

- 1.

IPA 서버 탭에서 **Role-Based Access Control** (역할 기반 액세스 제어) 하위 메뉴를 열고 셀프 서비스 권한을 선택합니다.

2.

셀프 서비스 액세스 규칙 목록의 오른쪽 상단에 있는 **Add** 를 클릭합니다.



3.

셀프 서비스 권한 추가 창이 열립니다. 셀프 서비스 이름 필드에 새 셀프 서비스 규칙의 이름을 입력합니다. 공백은 사용할 수 있습니다.

**Add Self Service Permission**

Self-service \*  
name Adding Personal Info

Attributes \*  
Filter

|                                                   |                                                 |
|---------------------------------------------------|-------------------------------------------------|
| <input type="checkbox"/> audio                    | <input type="checkbox"/> businesscategory       |
| <input type="checkbox"/> carlicense               | <input type="checkbox"/> cn                     |
| <input type="checkbox"/> departmentnumber         | <input type="checkbox"/> description            |
| <input type="checkbox"/> homedirectory            | <input type="checkbox"/> homephone              |
| <input type="checkbox"/> homepostaladdress        | <input type="checkbox"/> inetuserhttpurl        |
| <input type="checkbox"/> inetuserstatus           | <input checked="" type="checkbox"/> initials    |
| <input type="checkbox"/> internationalisdnumber   | <input type="checkbox"/> ipasshpubkey           |
| <input type="checkbox"/> ipatokenradiusconfiglink | <input type="checkbox"/> ipatokenradiususername |
| <input type="checkbox"/> ipauniqueid              | <input type="checkbox"/> ipauserauthtype        |
| <input checked="" type="checkbox"/> jpegphoto     | <input type="checkbox"/> krbcanonicalname       |

\* Required field

4. 사용자가 편집할 수 있는 속성 옆에 있는 확인란을 선택합니다.
5. **선택 사항:** 액세스 권한을 제공하려는 속성이 목록에 없는 경우 목록을 추가할 수 있습니다.
  - a. 추가 버튼을 클릭합니다.
  - b. 다음 **Add Custom Attribute** (사용자 지정 속성 추가) 창의 **Attribute** 텍스트 필드에 속성 이름을 입력합니다.
  - c. 확인 버튼을 클릭하여 특성을 추가합니다.
  - d. 새 속성이 선택되었는지 확인합니다.
6. 양식 하단의 **Add** (추가) 버튼을 클릭하여 새 셀프 서비스 규칙을 저장합니다.  
또는 추가 버튼을 클릭하고 다른 버튼을 추가하여 셀프 서비스 규칙을 저장하고 편집할 수 있습니다.

니다.

### 9.3. IDM 웹 UI를 사용하여 셀프 서비스 규칙 편집

이 절차에서는 웹 인터페이스(IdM Web UI)를 사용하여 IdM에서 셀프 서비스 액세스 규칙을 편집하는 방법을 설명합니다.

#### 사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.

#### 절차

1. IPA 서버 탭에서 **Role-Based Access Control** (역할 기반 액세스 제어) 하위 메뉴를 열고 셀프 서비스 권한을 선택합니다.
2. 수정할 셀프 서비스 규칙의 이름을 클릭합니다.

Self Service Permissions » User Self service

## Self Service Permission: User Self service

**Settings**

[Refresh](#) [Reset](#) [Update](#)

### General

Self-service name      User Self service

Attributes \*  [Add](#)

|                                                              |                                                      |
|--------------------------------------------------------------|------------------------------------------------------|
| <input type="checkbox"/> audio                               | <input checked="" type="checkbox"/> businesscategory |
| <input checked="" type="checkbox"/> carlicense               | <input checked="" type="checkbox"/> cn               |
| <input type="checkbox"/> departmentnumber                    | <input checked="" type="checkbox"/> description      |
| <input type="checkbox"/> destinationindicator                | <input checked="" type="checkbox"/> displayname      |
| <input type="checkbox"/> employeenumber                      | <input checked="" type="checkbox"/> employeetype     |
| <input checked="" type="checkbox"/> facsimiletelephonenumber | <input checked="" type="checkbox"/> gecos            |
| <input type="checkbox"/> gidnumber                           | <input checked="" type="checkbox"/> givenname        |
| <input type="checkbox"/> homedirectory                       | <input checked="" type="checkbox"/> homephone        |
| <input type="checkbox"/> homepostaladdress                   | <input checked="" type="checkbox"/> inetuserhttpurl  |
| <input type="checkbox"/> inetuserstatus                      | <input checked="" type="checkbox"/> initials         |

3.

편집 페이지에서만 셀프 서비스 규칙에 추가하거나 제거할 속성 목록을 편집할 수 있습니다. 적절한 확인란을 선택하거나 선택 취소합니다.

4.

저장 버튼을 클릭하여 변경 사항을 셀프 서비스 규칙에 저장합니다.

#### 9.4. IDM 웹 UI를 사용하여 셀프 서비스 규칙 삭제

이 절차에서는 웹 인터페이스(IdM Web UI)를 사용하여 IdM에서 셀프 서비스 액세스 규칙을 삭제하는 방법을 설명합니다.

##### 사전 요구 사항

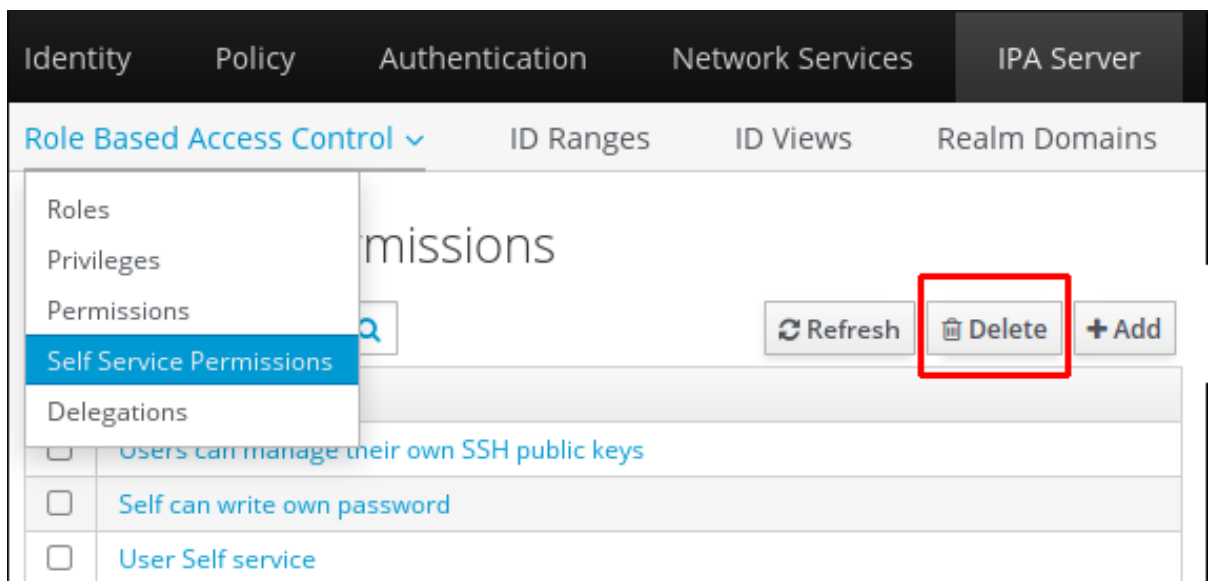
•

IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

- **IdM 웹 UI에 로그인되어 있습니다.** 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스를 참조 하십시오.](#)

## 절차

1. **IPA 서버 탭에서 Role-Based Access Control (역할 기반 액세스 제어) 하위 메뉴를 열고 셀프 서비스 권한을 선택합니다.**
2. 삭제할 규칙 옆에 있는 확인란을 선택한 다음 목록 오른쪽에 있는 삭제 버튼을 클릭합니다.



3. 대화 상자가 열리면 삭제를 클릭하여 확인합니다.

## 10장. ANSIBLE 플레이북을 사용하여 IDM에서 셀프 서비스 규칙 관리

이 섹션에서는 **IdM(Identity Management)**에 셀프 서비스 규칙을 소개하고 **Ansible** 플레이북을 사용하여 셀프 서비스 액세스 규칙을 만들고 편집하는 방법을 설명합니다. 셀프 서비스 액세스 제어 규칙을 사용하면 **IdM** 엔티티에서 **IdM Directory Server** 항목에서 지정된 작업을 수행할 수 있습니다.

이 섹션에서는 다음 주제를 다룹니다.

- [IdM의 셀프 서비스 액세스 제어](#)
- [Ansible을 사용하여 셀프 서비스 규칙이 있는지 확인합니다.](#)
- [Ansible을 사용하여 셀프 서비스 규칙이 없는지 확인](#)
- [Ansible을 사용하여 셀프 서비스 규칙에 특정 속성이 있는지 확인합니다.](#)
- [Ansible을 사용하여 셀프 서비스 규칙에 특정 속성이 없는지 확인합니다.](#)

### 10.1. IDM의 셀프 서비스 액세스 제어

셀프 서비스 액세스 제어 규칙은 **IdM** 디렉터리 서버 항목에서 수행할 수 있는 **IdM(Identity Management)** 엔티티의 작업을 정의합니다. 예를 들어 **IdM** 사용자는 자체 암호를 업데이트할 수 있습니다.

이 제어 방법을 사용하면 인증된 **IdM** 엔티티가 **LDAP** 항목 내에서 특정 속성을 편집할 수 있지만 전체 항목에 대한 작업 추가 또는 삭제는 허용하지 않습니다.



### 주의

셀프 서비스 액세스 제어 규칙으로 작업할 때는 주의해야 합니다. 액세스 제어 규칙을 부적절하게 구성하면 엔터티의 권한이 실수로 상승할 수 있습니다.

## 10.2. ANSIBLE을 사용하여 셀프 서비스 규칙이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 셀프 서비스 규칙을 정의하고 **IdM(Identity Management)** 서버에 있는지 확인하는 방법을 설명합니다. 이 예에서 새 사용자는 고유한 이름 세부 정보 규칙을 관리할 수 있으므로 사용자에게 자신의 지정된 이름, 표시 이름, 제목 및 초기 속성을 변경할 수 있습니다. 예를 들어 표시 이름 또는 초기 설정을 원하는 경우 변경할 수 있습니다.

### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - `~/MyPlaybooks/` 디렉터리에서 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

### 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/selfservice/` 디렉터리에 있는 **selfservice-present.yml** 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-present.yml
selfservice-present-copy.yml
```

3.

편집할 **selfservice-present-copy.yml** Ansible 플레이북 파일을 엽니다.

4.

**ipaselfservice** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 IdM 관리자의 암호로 설정합니다.
- **name** 변수를 새 셀프 서비스 규칙의 이름으로 설정합니다.
- 권한 변수를 쉼표로 구분된 권한 목록으로 설정하여 부여할 권한( 읽기 및 쓰기 )을 부여합니다.
- 특성 변수를 사용자가 직접 관리할 수 있는 속성 목록( **givenname,displayname,title,initials** )으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Self-service present
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure self-service rule "Users can manage their own name details" is
      present
      ipaselfservice:
        ipaadmin_password: Secret123
        name: "Users can manage their own name details"
        permission: read, write
        attribute:
          - givenname
          - displayname
          - title
          - initials
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory selfservice-present-copy.yml
```

추가 리소스

- [IdM의 셀프 서비스 액세스 제어를](#) 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-selfservice.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/selfservice` 디렉토리를 참조하십시오.

### 10.3. ANSIBLE을 사용하여 셀프 서비스 규칙이 없는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 구성에 지정된 셀프 서비스 규칙이 없는지 확인하는 방법을 설명합니다. 아래 예제에서는 사용자가 **IdM**에 자체 이름 세부 정보 셀프 서비스 규칙이 없는지 확인하는 방법을 설명합니다. 예를 들어 사용자가 자신의 표시 이름 또는 초기 이름을 변경할 수 없습니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - `~/MyPlaybooks/` 디렉터리에서 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/selfservice/` 디렉터리에 있는 `selfservice-absent.yml` 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-absent.yml  
selfservice-absent-copy.yml
```

3. 편집할 `selfservice-absent-copy.yml` Ansible 플레이북 파일을 엽니다.

4. `ipaselfservice` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- `name` 변수를 셀프 서비스 규칙의 이름으로 설정합니다.
- `state` 변수를 `absent` 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---  
- name: Self-service absent  
  hosts: ipaserver  
  become: true  
  
  tasks:  
    - name: Ensure self-service rule "Users can manage their own name details" is  
      absent  
      ipaselfservice:  
        ipaadmin_password: Secret123  
        name: "Users can manage their own name details"  
        state: absent
```

5. 파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory selfservice-absent-copy.yml
```

추가 리소스

- [IdM의 셀프 서비스 액세스 제어를](#) 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-selfservice.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/selfservice` 디렉터리에서 샘플 플레이북을 참조하십시오.

#### 10.4. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 속성이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 기존 셀프 서비스 규칙에 특정 설정이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 사용자가 자체 이름 세부 정보 **self-service** 규칙에 **surname member** 속성도 관리할 수 있는지 확인합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - `~/MyPlaybooks/` 디렉터리에서 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

- 사용자는 **IdM**에 자체 이름 세부 정보 셀프 서비스 규칙이 있을 수 있습니다.

## 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/selfservice/` 디렉터리에 있는 **selfservice-member-present.yml** 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-member-present.yml selfservice-member-present-copy.yml
```

3. 편집할 **selfservice-member-present-copy.yml** **Ansible** 플레이북 파일을 엽니다.
4. **ipaselfservice** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 수정할 셀프 서비스 규칙의 이름으로 설정합니다.
- 특성 변수를 **surname** 으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Self-service member present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure selfservice "Users can manage their own name details" member
```

```

attribute surname is present
ipaselfservice:
  ipaadmin_password: Secret123
  name: "Users can manage their own name details"
  attribute:
    - surname
  action: member

```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory selfservice-member-present-copy.yml
```

#### 추가 리소스

- [IdM의 셀프 서비스 액세스 제어를](#) 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리에서 사용 가능한 **README-selfservice.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/selfservice` 디렉토리에서 샘플 플레이북을 참조하십시오.

### 10.5. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 속성이 없는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 셀프 서비스 규칙에 특정 설정이 없는지 확인하는 방법을 설명합니다. 이 플레이북을 사용하여 셀프 서비스 규칙이 바람직하지 않은 액세스 권한을 부여하지 않도록 할 수 있습니다. 이 예제에서는 사용자가 자체 이름 세부 정보 **self-service** 규칙에 **givenname** 및 **surname member** 속성이 없는지 확인합니다.

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.

- **Ansible** 버전 **2.8** 이상을 사용하고 있습니다.
- **ansible-freeipa** 패키지를 설치했습니다.
- `~/MyPlaybooks/` 디렉터리에서 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- 사용자는 **IdM**에 자체 이름 세부 정보 셀프 서비스 규칙이 있을 수 있습니다.

## 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/selfservice/` 디렉터리에 있는 **selfservice-member-absent.yml** 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-member-absent.yml selfservice-member-absent-copy.yml
```

3. 편집할 **selfservice-member-absent-copy.yml** **Ansible** 플레이북 파일을 엽니다.

4. **ipaselfservice** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 수정하려는 셀프 서비스 규칙의 이름으로 설정합니다.
- 특정 변수를 **givenname** 및 **surname** 으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

- **state** 변수를 **absent** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Self-service member absent
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure selfservice "Users can manage their own name details" member
      attributes givenname and surname are absent
      ipaselfservice:
        ipaadmin_password: Secret123
        name: "Users can manage their own name details"
        attribute:
          - givenname
          - surname
        action: member
        state: absent
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory selfservice-member-absent-copy.yml
```

추가 리소스

- [IdM의 셀프 서비스 액세스 제어를](#) 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-selfservice.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/selfservice` 디렉터리에서 샘플 플레이북을 참조하십시오.

## 11장. IDM CLI에서 사용자 그룹 관리

이 장에서는 **IdM CLI**를 사용한 사용자 그룹 관리를 소개합니다.

사용자 그룹은 공통 권한, 암호 정책 및 기타 특성을 가진 사용자 집합입니다.

**IdM(Identity Management)**의 사용자 그룹에는 다음이 포함될 수 있습니다.

- **IdM 사용자**
- **기타 IdM 사용자 그룹**
- **외부 사용자: IdM 외부에 있는 사용자**

### 11.1. IDM의 다양한 그룹 유형

**IdM**은 다음 유형의 그룹을 지원합니다.

#### **POSIX 그룹(기본값)**

**POSIX** 그룹은 해당 멤버에 대해 **Linux POSIX** 속성을 지원합니다. **Active Directory**와 상호 작용하는 그룹에서는 **POSIX** 속성을 사용할 수 없습니다.

**POSIX** 속성은 사용자를 별도의 엔티티로 식별합니다. 사용자와 관련된 **POSIX** 속성의 예로는 **uidNumber**, **UID**(사용자 번호), **gidNumber**, 그룹 번호(**GID**)가 있습니다.

#### **비POSIX 그룹**

**비POSIX** 그룹은 **POSIX** 속성을 지원하지 않습니다. 예를 들어 이러한 그룹에는 **GID**가 정의되어 있지 않습니다.

이러한 유형의 그룹의 모든 멤버가 **IdM** 도메인에 속해야 합니다.

#### **외부 그룹**

외부 그룹을 사용하여 **IdM** 도메인 외부의 **ID** 저장소에 존재하는 그룹 멤버를 다음과 같이 추가합니다.

- 로컬 시스템
- **Active Directory** 도메인
- 디렉터리 서비스

외부 그룹은 **POSIX** 속성을 지원하지 않습니다. 예를 들어 이러한 그룹에는 **GID**가 정의되어 있지 않습니다.

표 11.1. 기본적으로 생성된 사용자 그룹

| 그룹 이름           | 기본 그룹 멤버                                   |
|-----------------|--------------------------------------------|
| <b>ipausers</b> | 모든 IdM 사용자                                 |
| 관리자             | 기본 <b>admin</b> 사용자를 포함하여 관리자 권한이 있는 사용자   |
| <b>editors</b>  | 이는 더 이상 특별한 권한이 없는 레거시 그룹입니다.              |
| 신뢰 관리자          | Active Directory 트러스트를 관리할 수 있는 권한이 있는 사용자 |

사용자 그룹에 사용자를 추가하면 사용자와 연결된 권한과 정책이 제공됩니다. 예를 들어 사용자에게 관리 권한을 부여하려면 사용자를 **admins** 그룹에 추가합니다.



#### 주의

**admins** 그룹을 삭제하지 마십시오. 관리자는 **IdM**에 필요한 사전 정의된 그룹이므로 이 작업으로 인해 특정 명령에 문제가 발생합니다.

또한 **IdM**은 **IdM**에 새 사용자를 생성할 때마다 기본적으로 사용자 개인 그룹을 생성합니다. 개인 그룹에 대한 자세한 내용은 개인 그룹 없이 사용자 추가를 참조하십시오.

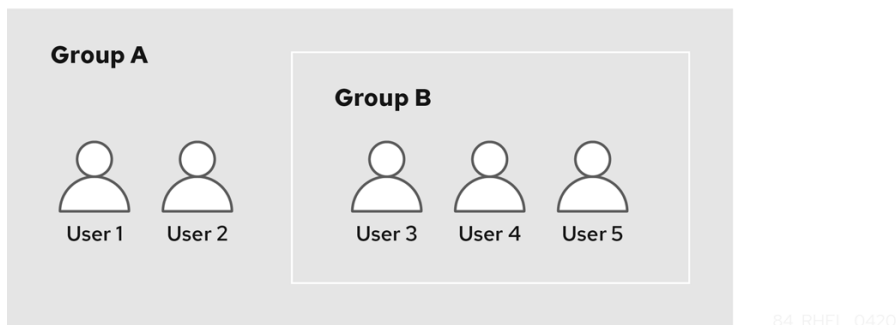
## 11.2. 직접 및 간접 그룹 멤버

IdM의 사용자 그룹 속성은 직접 및 간접 멤버 모두에 적용됩니다. 그룹 B가 A 그룹의 멤버인 경우 B 그룹의 모든 사용자는 A 그룹의 간접 멤버로 간주됩니다.

예를 들어 다음 다이어그램에서: For example, in the following diagram:

- 사용자 1과 사용자 2는 A 그룹의 직접적인 멤버입니다.
- 사용자 3, 사용자 4 및 사용자 5는 A 그룹의 간접 멤버입니다.

그림 11.1. 직접 및 간접 그룹 멤버십



B4\_RHEL\_0420

사용자 그룹 A에 대한 암호 정책을 설정하는 경우 정책은 사용자 그룹 B의 모든 사용자에게도 적용됩니다.

## 11.3. IDM CLI를 사용하여 사용자 그룹 추가

이 섹션에서는 IdM CLI를 사용하여 사용자 그룹을 추가하는 방법을 설명합니다.

### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#)을 참조하십시오.

### 절차

- **ipa group-add *group\_name*** 명령을 사용하여 사용자 그룹을 추가합니다. 예를 들어 **group\_a**를 생성하려면 다음을 수행합니다.

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

기본적으로 **ipa group-add** 는 **POSIX** 사용자 그룹을 추가합니다. 다른 그룹 유형을 지정하려면 **ipa group-add** 에 옵션을 추가합니다.

- **--nonposix**: 비POSIX 그룹 생성
- 외부 그룹을 생성하는 **--external**

그룹 유형에 대한 자세한 내용은 **IdM의 다른 그룹 유형**을 참조하십시오.

**--gid=*custom\_GID*** 옵션을 사용하여 사용자 그룹을 추가할 때 사용자 정의 **GID**를 지정 할 수 있습니다. 이렇게 하는 경우 **ID** 충돌을 피하기 위해 주의하십시오. 사용자 정의 **GID**를 지정하지 않으면 **IdM**에서 사용 가능한 **ID** 범위에서 **GID**를 자동으로 할당합니다.



#### 주의

로컬 그룹을 **IdM**에 추가하지 마십시오. **NSS(Name Service Switch)**는 로컬 사용자 및 그룹을 확인하기 전에 항상 **IdM** 사용자 및 그룹을 확인합니다. 즉, 예를 들어 **IdM** 그룹 멤버십이 로컬 사용자에게는 작동하지 않습니다.

## 11.4. IDM CLI를 사용하여 사용자 그룹 검색

이 섹션에서는 **IdM CLI**를 사용하여 기존 사용자 그룹을 검색하는 방법을 설명합니다.

절차

- **ipa group-find** 명령을 사용하여 모든 사용자 그룹을 표시합니다. 그룹 유형을 지정하려면 **ipa group-find** 에 옵션을 추가합니다.
    - **ipa group-find --posix** 명령을 사용하여 모든 **POSIX** 그룹을 표시합니다.
    - **ipa group-find --nonposix** 명령을 사용하여 비**POSIX** 그룹을 모두 표시합니다.
    - **ipa group-find --external** 명령을 사용하여 모든 외부 그룹을 표시합니다.
- 다양한 그룹 유형에 대한 자세한 내용은 [IdM의 다른 그룹 유형](#)을 참조하십시오.

## 11.5. IDM CLI를 사용하여 사용자 그룹 삭제

이 섹션에서는 **IdM CLI**를 사용하여 사용자 그룹을 삭제하는 방법을 설명합니다. 그룹을 삭제해도 **IdM**에서 그룹 멤버가 삭제되지 않습니다.

### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#)을 참조하십시오.

### 절차

- **ipa group-del group\_name** 명령을 사용하여 사용자 그룹을 삭제합니다. 예를 들어 **group\_a**를 삭제하려면 다음을 수행합니다.

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

## 11.6. IDM CLI를 사용하여 사용자 그룹에 멤버 추가

이 섹션에서는 **IdM CLI**를 사용하여 사용자 그룹에 멤버를 추가하는 방법을 설명합니다. 사용자와 사용자 그룹을 모두 사용자 그룹의 멤버로 추가할 수 있습니다. 자세한 내용은 [IdM 및 직접 및 간접 그룹 멤버의 다른 그룹 유형](#)을 참조하십시오.

## 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#)을 참조하십시오.

## 절차

- `ipa group-add-member` 명령을 사용하여 사용자 그룹에 멤버를 추가합니다.
- 다음 옵션을 사용하여 멤버 유형을 지정합니다.
- `--users` 에서 IdM 사용자 추가
  - `--external` 은 IdM 도메인 외부에 존재하는 사용자를 `DOMAIN\user_name` 또는 `user_name@domain` 형식으로 추가합니다.
  - `--groups` 에서 IdM 사용자 그룹 추가

예를 들어 `group_b`를 `group_a`의 멤버로 추가하려면 다음을 수행합니다.

```
$ 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`의 멤버는 이제 `group_a`의 간접 멤버입니다.

## 중요

그룹을 다른 그룹의 멤버로 추가할 때 채워 그룹을 만들지 마십시오. 예를 들어 **Group A**가 **Group B**의 멤버인 경우 **그룹 B**를 **그룹 A**의 멤버로 추가하지 마십시오. 반복 그룹은 예기치 않은 동작이 발생할 수 있습니다.



## 참고

사용자 그룹에 멤버를 추가한 후 업데이트에는 ID 관리 환경의 모든 클라이언트에 분산되는 데 시간이 걸릴 수 있습니다. 이는 지정된 호스트가 사용자, 그룹 및 네트워크 그룹을 확인할 때 **SSSD( System Security Services Daemon )**가 먼저 캐시를 조회하고 누락되거나 만료된 레코드에 대해서만 서버 조회를 수행하기 때문입니다.

## 11.7. 사용자 개인 그룹 없이 사용자 추가

기본적으로 **IdM**은 새 사용자가 **IdM**에 생성될 때마다 사용자 개인 그룹(**UPG**)을 생성합니다. **UPGs**는 특정 그룹 유형입니다.

- **UPG**는 새로 생성된 사용자와 동일한 이름을 갖습니다.
- 사용자는 **UPG**의 유일한 멤버입니다. **UPG**는 다른 멤버를 포함할 수 없습니다.
- 개인 그룹의 **GID**는 사용자의 **UID**와 일치합니다.

그러나 **UPG**를 생성하지 않고 사용자를 추가할 수 있습니다.

### 11.7.1. 사용자 개인 그룹이 없는 사용자

**NIS** 그룹 또는 다른 시스템 그룹이 사용자 개인 그룹에 할당되는 **GID**를 이미 사용하는 경우 **UPG**를 생성하지 않아야 합니다.

이 작업은 다음 두 가지 방법으로 수행할 수 있습니다.

- 전역적으로 개인 그룹을 비활성화하지 않고 **UPG** 없이 새 사용자를 추가합니다. 개인 그룹이 전역적으로 활성화된 경우 사용자 개인 그룹이 없는 사용자 추가 를 참조하십시오.
- 모든 사용자에게 전역적으로 **UPG**를 비활성화한 다음 새 사용자를 추가합니다. 모든 사용자에게 전역적으로 사용자 개인 그룹 비활성화 및 사용자 개인 그룹이 전역적으로 비활성화될 때 사용자 추가 를 참조하십시오.

두 경우 모두 **IdM**에서 새 사용자를 추가할 때 **GID**를 지정해야 합니다. 그렇지 않으면 작업이 실패합니다.

다. **IdM**에는 새 사용자에 대한 **GID**가 필요하지만 기본 사용자 그룹 **ipausers**는 **POSIX** 그룹이 아니므로 연결된 **GID**가 없기 때문입니다. 지정한 **GID**는 이미 존재하는 그룹에 해당할 필요가 없습니다.



#### 참고

**GID**를 지정하면 새 그룹이 생성되지 않습니다. **IdM**에서 속성이 필요하므로 새 사용자에게 대한 **GID** 속성만 설정합니다.

### 11.7.2. 개인 그룹이 전역적으로 활성화된 경우 사용자 개인 그룹이 없는 사용자 추가

시스템에서 **UPG**를 사용하도록 설정한 경우에도 사용자 개인 그룹(**UPG**)을 생성하지 않고 사용자를 추가할 수 있습니다. 이를 위해서는 새 사용자에 대한 **GID**를 수동으로 설정해야 합니다. 필요한 이유에 대한 자세한 내용은 [사용자 개인 그룹이 없는 사용자](#)를 참조하십시오.

#### 절차



**IdM**이 **UPG**를 생성하지 못하도록 **ipa user-add** 명령에 **--noprivate** 옵션을 추가합니다.

명령이 성공하려면 사용자 지정 **GID**를 지정해야 합니다. 예를 들어 **GID 10000**이 있는 새 사용자를 추가하려면 다음을 수행합니다.

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

### 11.7.3. 모든 사용자에게 대해 사용자 개인 그룹 비활성화

전역적으로 사용자 개인 그룹(**UPG**)을 비활성화할 수 있습니다. 이렇게 하면 모든 새 사용자에게 대한 **UPG**가 생성되지 않습니다. 기존 사용자는 이 변경 사항의 영향을 받지 않습니다.

#### 절차

1.

관리자 권한을 얻습니다.

```
$ kinit admin
```

2.

**IdM**은 **Directory Server Managed Entries** 플러그인을 사용하여 **UPG**를 관리합니다. 플러그인의 인스턴스를 나열합니다.

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

3.

IdM이 UPGs를 생성하지 않도록 하려면 사용자 개인 그룹 관리를 담당하는 플러그인 인스턴스를 비활성화합니다.

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

Disabling Plugin



참고

나중에 UPG 정의 인스턴스를 다시 사용하려면 `ipa-managed-entries -e "UPG Definition" enable` 명령을 사용합니다.

4.

Directory Server를 다시 시작하여 새 구성을 로드합니다.

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

UPGs를 비활성화한 후 사용자를 추가하려면 GID를 지정해야 합니다. 자세한 내용은 사용자 개인 그룹이 전역적으로 비활성화될 때 사용자 추가를 참조하십시오.

#### 검증 단계

•

UPG가 전역적으로 비활성화되었는지 확인하려면 `disable` 명령을 다시 사용하십시오.

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

Plugin already disabled

#### 11.7.4. 사용자 개인 그룹이 전역적으로 비활성화될 때 사용자 추가

사용자 개인 그룹(UPG)이 전역적으로 비활성화되면 IdM에서 새 사용자에게 GID를 자동으로 할당하지 않습니다. 사용자를 추가하려면 수동으로 또는 `automember` 규칙을 사용하여 GID를 할당해야 합니다. 필요한 이유에 대한 자세한 내용은 사용자 개인 그룹이 없는 사용자를 참조하십시오.

#### 전제 조건

•

UPG는 모든 사용자에게 대해 전역적으로 비활성화해야 합니다. 자세한 내용은 모든 사용자에게 대해 전역 사용자 개인 그룹 비활성화를 참조하십시오.

#### 절차

- **UPGs를 생성할 때 새 사용자를 추가하는 데 성공하려면 다음 중 하나를 선택합니다.**
  - 새 사용자를 추가할 때 사용자 정의 **GID**를 지정합니다. **GID**는 이미 존재하는 사용자 그룹에 해당할 필요가 없습니다.

예를 들어 명령줄에서 사용자를 추가할 때 **ipa user-add** 명령에 **--gid** 옵션을 추가합니다.
  - **automember** 규칙을 사용하여 **GID**가 있는 기존 그룹에 사용자를 추가합니다.

## 11.8. IDM CLI를 사용하여 IDM 사용자 그룹에 사용자 또는 그룹 추가

이 섹션에서는 **IdM CLI**를 사용하여 사용자 또는 그룹을 멤버 관리자로 **IdM** 사용자 그룹에 추가하는 방법을 설명합니다. 멤버 관리자는 **IdM** 사용자 그룹에 사용자 또는 그룹을 추가할 수 있지만 그룹의 속성은 변경할 수 없습니다.

### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#)을 참조하십시오.
- 멤버 관리자로 추가하는 사용자 또는 그룹의 이름과 관리할 그룹의 이름이 있어야 합니다.

### 절차

- **ipa group-add-member-manager** 명령을 사용하여 사용자를 **IdM** 사용자 그룹에 추가합니다.

예를 들어 사용자 **test** 를 **group\_a** 의 멤버 관리자로 추가하려면 다음을 수행합니다.

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

사용자 테스트는 **group\_a**의 멤버를 관리할 수 있습니다.

- **ipa group-add-member-manager** 명령을 사용하여 그룹 관리자를 **IdM** 사용자 그룹에 추가합니다.

예를 들어 **group\_admins** 그룹을 **group\_a**의 멤버 관리자로 추가하려면 다음을 수행합니다.

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

그룹 **group\_admins**는 **group\_a**의 멤버를 관리할 수 있습니다.



#### 참고

사용자 그룹에 멤버 관리자를 추가한 후 업데이트에는 **ID** 관리 환경의 모든 클라이언트에 분산되는 데 시간이 걸릴 수 있습니다.

#### 검증 단계

- **ipa group-show** 명령을 사용하여 사용자와 그룹이 멤버 관리자로 추가되었는지 확인합니다.

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

#### 추가 리소스

- 자세한 내용은 **ipa group-add-member-manager --help**를 참조하십시오.

## 11.9. IDM CLI를 사용하여 그룹 멤버 보기

이 섹션에서는 **IdM CLI**를 사용하여 그룹의 멤버를 보는 방법에 대해 설명합니다. 직접 및 간접 그룹 멤

버를 모두 볼 수 있습니다. 자세한 내용은 [직접 및 간접 그룹 멤버를 참조하십시오](#).

절차:

- 그룹의 멤버를 나열하려면 `ipa group-show group_name` 명령을 사용합니다. 예를 들어 다음과 같습니다.

```
$ ipa group-show group_a
...
Member users: user_a
Member groups: group_b
Indirect Member users: user_b
```



참고

간접 멤버 목록에는 신뢰할 수 있는 **Active Directory** 도메인의 외부 사용자가 포함되어 있지 않습니다. **Active Directory** 신뢰 사용자 오브젝트는 **ID** 관리 내에 **LDAP** 개체로 존재하지 않기 때문에 **ID** 관리 인터페이스에 표시되지 않습니다.

#### 11.10. IDM CLI를 사용하여 사용자 그룹에서 멤버 제거

이 섹션에서는 **IdM CLI**를 사용하여 사용자 그룹에서 멤버를 제거하는 방법을 설명합니다.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#)을 참조하십시오.

절차

1. **선택 사항:** `ipa group-show` 명령을 사용하여 그룹에 제거하려는 멤버가 포함되어 있는지 확인합니다.
2. `ipa group-remove-member` 명령을 사용하여 사용자 그룹에서 멤버를 제거합니다.

다음 옵션을 사용하여 제거할 멤버를 지정합니다.

- `--users` 에서 **IdM** 사용자 제거

- **--external** 은 IdM 도메인 외부에 존재하는 사용자를 **DOMAIN\user\_name** 또는 **user\_name@domain** 형식으로 제거합니다.
- **--groups** 에서 IdM 사용자 그룹 제거

예를 들어 **group\_name** 이라는 그룹에서 **user1,user2** 및 **group1** 을 제거하려면 다음을 수행합니다.

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

### 11.11. IDM CLI를 사용하여 IDM 사용자 그룹에서 멤버 관리자로 사용자 또는 그룹 제거

이 섹션에서는 IdM CLI를 사용하여 IdM 사용자 그룹에서 멤버 관리자로 사용자 또는 그룹을 제거하는 방법을 설명합니다. 멤버 관리자는 IdM 사용자 그룹에서 사용자 또는 그룹을 제거할 수 있지만 그룹의 속성은 변경할 수 없습니다.

#### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#) 을 참조하십시오.
- 제거 중인 기존 멤버 관리자 사용자 또는 그룹의 이름과 관리 중인 그룹의 이름이 있어야 합니다.

#### 절차

- **ipa group-remove-member-manager** 명령을 사용하여 IdM 사용자 그룹의 멤버 관리자로 사용자를 제거합니다.

예를 들어 사용자 **test** 를 **group\_a** 의 멤버 관리자로 제거하려면 다음을 수행합니다.

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

사용자 테스트 는 더 이상 **group\_a** 의 멤버를 관리할 수 없습니다.

- **ipa group-remove-member-manager** 명령을 사용하여 **IdM** 사용자 그룹의 멤버 관리자로 그룹을 제거합니다.

예를 들어 **group\_admins** 그룹을 **group\_a** 의 멤버 관리자로 제거하려면 다음을 수행합니다.

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

**group group\_admins** 는 **group\_a** 의 멤버를 더 이상 관리할 수 없습니다.



#### 참고

사용자 그룹에서 멤버 관리자를 제거한 후 업데이트는 **ID** 관리 환경의 모든 클라이언트에 전파되는 데 시간이 걸릴 수 있습니다.

#### 검증 단계

- **ipa group-show** 명령을 사용하여 사용자와 그룹이 멤버 관리자로 제거되었는지 확인합니다.

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

#### 추가 리소스

- 자세한 내용은 **ipa group-remove-member-manager --help** 를 참조하십시오.

## 12장. IDM 웹 UI에서 사용자 그룹 관리

이 장에서는 **IdM 웹 UI**를 사용한 사용자 그룹 관리를 소개합니다.

사용자 그룹은 공통 권한, 암호 정책 및 기타 특성을 가진 사용자 집합입니다.

**IdM(Identity Management)**의 사용자 그룹에는 다음이 포함될 수 있습니다.

- **IdM 사용자**
- **기타 IdM 사용자 그룹**
- **외부 사용자: IdM 외부에 있는 사용자**

### 12.1. IDM의 다양한 그룹 유형

**IdM**은 다음 유형의 그룹을 지원합니다.

#### POSIX 그룹(기본값)

**POSIX** 그룹은 해당 멤버에 대해 **Linux POSIX** 속성을 지원합니다. **Active Directory**와 상호 작용하는 그룹에서는 **POSIX** 속성을 사용할 수 없습니다.

**POSIX** 속성은 사용자를 별도의 엔티티로 식별합니다. 사용자와 관련된 **POSIX** 속성의 예로는 **uidNumber**, **UID**(사용자 번호), **gidNumber**, 그룹 번호(**GID**)가 있습니다.

#### 비POSIX 그룹

비**POSIX** 그룹은 **POSIX** 속성을 지원하지 않습니다. 예를 들어 이러한 그룹에는 **GID**가 정의되어 있지 않습니다.

이러한 유형의 그룹의 모든 멤버가 **IdM** 도메인에 속해야 합니다.

#### 외부 그룹

외부 그룹을 사용하여 **IdM** 도메인 외부의 **ID** 저장소에 존재하는 그룹 멤버를 다음과 같이 추가합니다.

- 로컬 시스템
- **Active Directory** 도메인
- 디렉터리 서비스

외부 그룹은 **POSIX** 속성을 지원하지 않습니다. 예를 들어 이러한 그룹에는 **GID**가 정의되어 있지 않습니다.

표 12.1. 기본적으로 생성된 사용자 그룹

| 그룹 이름           | 기본 그룹 멤버                                   |
|-----------------|--------------------------------------------|
| <b>ipausers</b> | 모든 IdM 사용자                                 |
| 관리자             | 기본 <b>admin</b> 사용자를 포함하여 관리자 권한이 있는 사용자   |
| <b>editors</b>  | 이는 더 이상 특별한 권한이 없는 레거시 그룹입니다.              |
| 신뢰 관리자          | Active Directory 트러스트를 관리할 수 있는 권한이 있는 사용자 |

사용자 그룹에 사용자를 추가하면 사용자와 연결된 권한과 정책이 제공됩니다. 예를 들어 사용자에게 관리 권한을 부여하려면 사용자를 **admins** 그룹에 추가합니다.



#### 주의

**admins** 그룹을 삭제하지 마십시오. 관리자는 **IdM**에 필요한 사전 정의된 그룹이므로 이 작업으로 인해 특정 명령에 문제가 발생합니다.

또한 **IdM**은 **IdM**에 새 사용자를 생성할 때마다 기본적으로 사용자 개인 그룹을 생성합니다. 개인 그룹에 대한 자세한 내용은 개인 그룹 [없이 사용자 추가](#)를 참조하십시오.

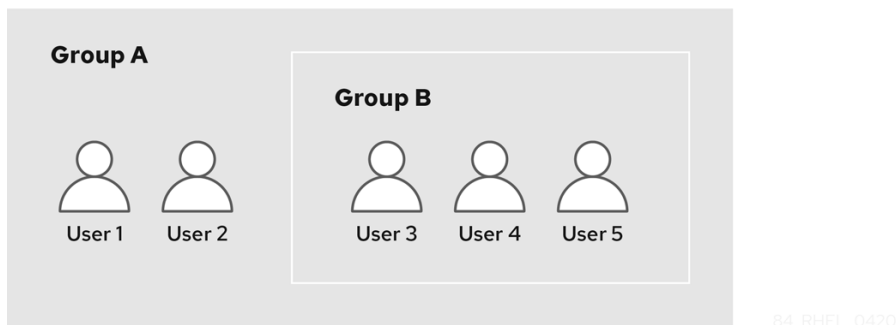
## 12.2. 직접 및 간접 그룹 멤버

IdM의 사용자 그룹 속성은 직접 및 간접 멤버 모두에 적용됩니다. 그룹 B가 A 그룹의 멤버인 경우 B 그룹의 모든 사용자는 A 그룹의 간접 멤버로 간주됩니다.

예를 들어 다음 다이어그램에서: For example, in the following diagram:

- 사용자 1과 사용자 2는 A 그룹의 직접적인 멤버입니다.
- 사용자 3, 사용자 4 및 사용자 5는 A 그룹의 간접 멤버입니다.

그림 12.1. 직접 및 직접 그룹 멤버십



B4\_RHEL\_0420

사용자 그룹 A에 대한 암호 정책을 설정하는 경우 정책은 사용자 그룹 B의 모든 사용자에게도 적용됩니다.

## 12.3. IDM 웹 UI를 사용하여 사용자 그룹 추가

이 섹션에서는 IdM 웹 UI를 사용하여 사용자 그룹을 추가하는 방법을 설명합니다.

### 사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.

### 절차

1. **ID** → 그룹 을 클릭하고 왼쪽 사이드바에서 사용자 그룹을 선택합니다.
2. 추가 를 클릭하여 그룹 추가를 시작합니다.
3. 그룹에 대한 정보를 입력합니다. 사용자 그룹 유형에 대한 자세한 내용은 **IdM의 다른 그룹 유형**을 참조하십시오.

그룹에 대한 사용자 정의 **GID**를 지정할 수 있습니다. 이렇게 하는 경우 **ID** 충돌을 피하기 위해 주의하십시오. 사용자 정의 **GID**를 지정하지 않으면 **IdM**에서 사용 가능한 **ID** 범위에서 **GID**를 자동으로 할당합니다.

**Add user group** [X]

Group name \*

Description

Group Type ☐ Non-POSIX ☐ External ☒ POSIX

GID

\* Required field

[Add] [Add and Add Another] [Add and Edit] [Cancel]

4. **Add** 를 클릭하여 확인합니다.

#### 12.4. IDM 웹 UI를 사용하여 사용자 그룹 삭제

이 섹션에서는 **IdM** 웹 UI를 사용하여 사용자 그룹을 삭제하는 방법을 설명합니다. 그룹을 삭제해도 **IdM**에서 그룹 멤버가 삭제되지 않습니다.

#### 사전 요구 사항

- **IdM 웹 UI에 로그인되어 있습니다.**

#### 절차

1. **ID** → **그룹**을 클릭하고 사용자 그룹을 선택합니다.
2. 삭제할 그룹을 선택합니다.
3. 삭제를 클릭합니다.
4. **Delete**를 클릭하여 확인합니다.

### 12.5. IDM 웹 UI를 사용하여 사용자 그룹에 멤버 추가

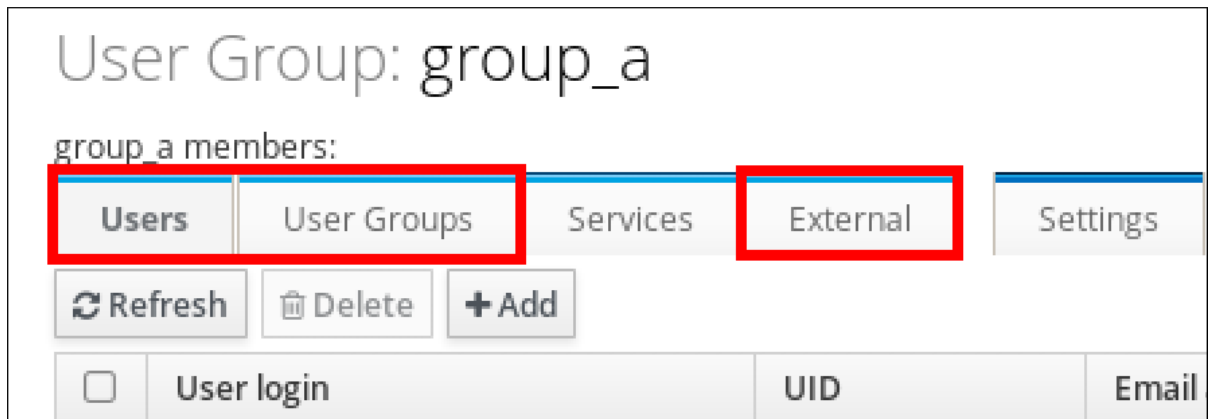
사용자와 사용자 그룹을 모두 사용자 그룹의 멤버로 추가할 수 있습니다. 자세한 내용은 [IdM 및 직접 및 간접 그룹 멤버의 다른 그룹 유형](#)을 참조하십시오.

#### 사전 요구 사항

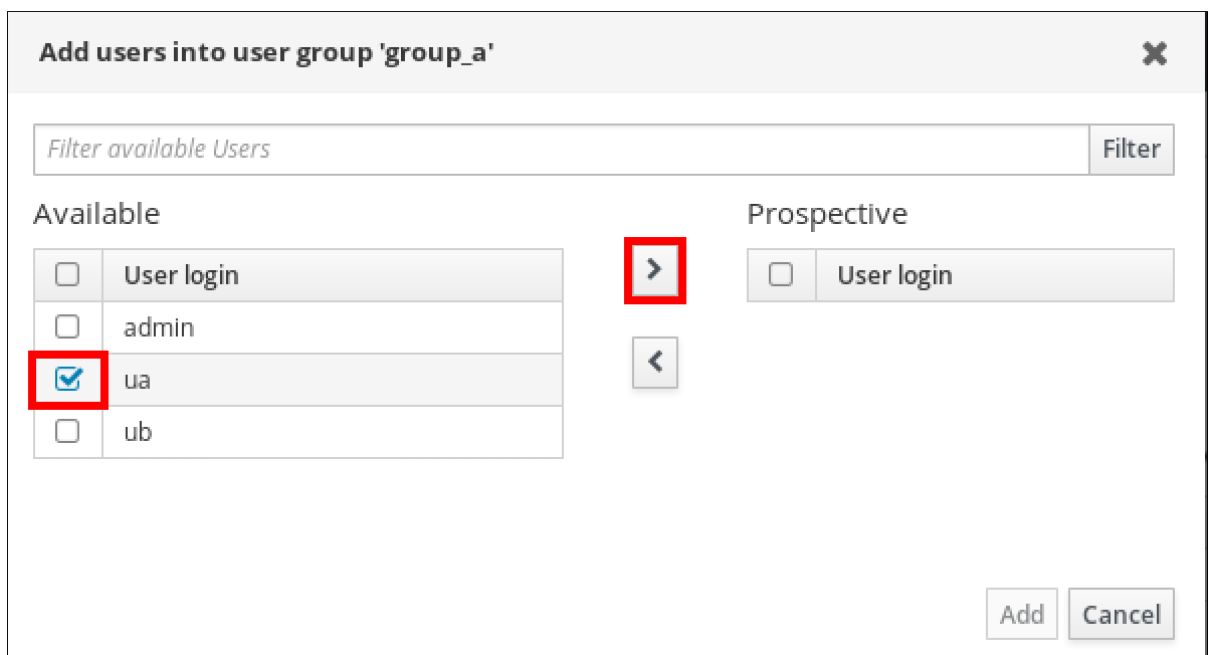
- **IdM 웹 UI에 로그인되어 있습니다.**

#### 절차

1. **ID** → **그룹**을 클릭하고 왼쪽 사이드바에서 사용자 그룹을 선택합니다.
2. 그룹 이름을 클릭합니다.
3. 추가할 그룹 멤버 유형을 선택합니다. 사용자, 사용자 그룹 또는 외부.



4. 추가를 클릭합니다.
5. 추가할 하나 이상의 멤버 옆에 있는 확인란을 선택합니다. **Select the check box next to one or more members you want to add.**
6. 오른쪽 화살표를 클릭하여 선택한 멤버를 그룹으로 이동합니다.



7. **Add** 를 클릭하여 확인합니다.

## 12.6. 웹 UI를 사용하여 IDM 사용자 그룹에 사용자 또는 그룹 추가

이 섹션에서는 웹 UI를 사용하여 사용자 또는 그룹을 멤버 관리자로 **IdM** 사용자 그룹에 추가하는 방법을 설명합니다. 멤버 관리자는 **IdM** 사용자 그룹에 사용자 또는 그룹을 추가할 수 있지만 그룹의 속성은 변경할 수 없습니다.

## 사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- 멤버 관리자로 추가하는 사용자 또는 그룹의 이름과 관리할 그룹의 이름이 있어야 합니다.

## 절차

1. ID → 그룹을 클릭하고 왼쪽 사이드바 에서 사용자 그룹을 선택합니다.
2. 그룹 이름을 클릭합니다.
3. 추가할 그룹 멤버 관리자 유형을 선택합니다. 사용자 또는 사용자 그룹.

## User Group: group\_a

group\_a members:

|       |             |          |          |                   |
|-------|-------------|----------|----------|-------------------|
| Users | User Groups | Services | External | User ID overrides |
|-------|-------------|----------|----------|-------------------|

group\_a member managers:
 

|             |       |
|-------------|-------|
| User Groups | Users |
|-------------|-------|

Refresh
Delete
+ Add

4. 추가를 클릭합니다.
5. 추가할 하나 이상의 멤버 옆에 있는 확인란을 선택합니다. **Select the check box next to one or more members you want to add.**
6. 오른쪽 화살표를 클릭하여 선택한 멤버를 그룹으로 이동합니다.

**Add users as member managers for user group 'group\_a'**
✕

Filter available Users
Filter

**Available**

|                                     |            |
|-------------------------------------|------------|
| <input type="checkbox"/>            | User login |
| <input type="checkbox"/>            | admin      |
| <input checked="" type="checkbox"/> | test1      |
| <input type="checkbox"/>            | test2      |
| <input type="checkbox"/>            | test_user  |
| <input type="checkbox"/>            | test_user2 |
| <input type="checkbox"/>            | tuser3     |

➤

➤

**Prospective**

|                          |            |
|--------------------------|------------|
| <input type="checkbox"/> | User login |
|--------------------------|------------|

Add

Cancel

7.

**Add** 를 클릭하여 확인합니다.



참고

사용자 그룹에 멤버 관리자를 추가한 후 업데이트에는 ID 관리 환경의 모든 클라이언트에 분산되는 데 시간이 걸릴 수 있습니다.

검증 단계

•

새로 추가된 사용자 또는 사용자 그룹이 사용자 또는 사용자 그룹의 멤버 관리자 목록에 추가되었는지 확인합니다.

## User Group: project

project members:

|       |             |          |
|-------|-------------|----------|
| Users | User Groups | Services |
|-------|-------------|----------|

project member managers:

|                 |       |
|-----------------|-------|
| User Groups (1) | Users |
|-----------------|-------|

|         |        |     |
|---------|--------|-----|
| Refresh | Delete | Add |
|---------|--------|-----|

|                          |                |
|--------------------------|----------------|
| <input type="checkbox"/> | Group name     |
| <input type="checkbox"/> | project_admins |

추가 리소스

- 자세한 내용은 `ipa group-add-member-manager --help` 를 참조하십시오.

### 12.7. IDM 웹 UI를 사용하여 그룹 멤버 보기

이 섹션에서는 IdM 웹 UI를 사용하여 그룹의 멤버를 보는 방법에 대해 설명합니다. 직접 및 간접 그룹 멤버를 모두 볼 수 있습니다. 자세한 내용은 [직접 및 간접 그룹 멤버](#)를 참조하십시오.

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.

절차

1. ID → 그룹을 선택합니다.
2. 왼쪽 사이드바 에서 사용자 그룹을 선택합니다.
3. 보려는 그룹의 이름을 클릭합니다.
4. 직접 멤버십과 직접 멤버십 을 전환합니다.

|                          |            |     |                                                                                                           |                  |           |  |
|--------------------------|------------|-----|-----------------------------------------------------------------------------------------------------------|------------------|-----------|--|
| Refresh                  | Delete     | Add | Show Results <input checked="" type="radio"/> Direct Membership <input type="radio"/> Indirect Membership |                  |           |  |
| <input type="checkbox"/> | User login | UID | Email address                                                                                             | Telephone Number | Job Title |  |

## 12.8. IDM 웹 UI를 사용하여 사용자 그룹에서 멤버 제거

이 섹션에서는 **IdM 웹 UI**를 사용하여 사용자 그룹에서 멤버를 제거하는 방법을 설명합니다.

사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.

절차

1. **ID** → 그룹을 클릭하고 왼쪽 사이드바 에서 사용자 그룹을 선택합니다.
2. 그룹 이름을 클릭합니다.
3. 제거할 그룹 멤버 유형을 선택합니다. 사용자, 사용자 그룹 또는 외부.

User Group: group\_a

group\_a members:

|              |             |          |                 |          |
|--------------|-------------|----------|-----------------|----------|
| <b>Users</b> | User Groups | Services | <b>External</b> | Settings |
|--------------|-------------|----------|-----------------|----------|

|         |        |     |
|---------|--------|-----|
| Refresh | Delete | Add |
|---------|--------|-----|

|                          |            |     |       |
|--------------------------|------------|-----|-------|
| <input type="checkbox"/> | User login | UID | Email |
|--------------------------|------------|-----|-------|

4. 제거할 멤버 옆에 있는 확인란을 선택합니다. **Select the check box next to the member you want to remove.**
5. 삭제를 클릭합니다.
6. **Delete** 를 클릭하여 확인합니다.

## 12.9. 웹 UI를 사용하여 IdM 사용자 그룹에서 멤버 관리자로 사용자 또는 그룹 제거

이 섹션에서는 웹 UI를 사용하여 IdM 사용자 그룹에서 사용자 또는 그룹을 멤버 관리자로 제거하는 방법을 설명합니다. 멤버 관리자는 IdM 사용자 그룹에서 사용자 또는 그룹을 제거할 수 있지만 그룹의 속성은 변경할 수 없습니다.

### 사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- 제거 중인 기존 멤버 관리자 사용자 또는 그룹의 이름과 관리 중인 그룹의 이름이 있어야 합니다.

### 절차

1. ID → 그룹을 클릭하고 왼쪽 사이드바에서 사용자 그룹을 선택합니다.
2. 그룹 이름을 클릭합니다.
3. 제거할 멤버 관리자 유형을 선택합니다. 사용자 또는 사용자 그룹.

User Group: group\_a

group\_a members:



4. 제거할 멤버 관리자 옆에 있는 확인란을 선택합니다. **Select the check box next to the member manager you want to remove.**
5. 삭제를 클릭합니다.

6.

**Delete** 를 클릭하여 확인합니다.



참고

사용자 그룹에서 멤버 관리자를 제거한 후 업데이트는 ID 관리 환경의 모든 클라이언트에 전파되는 데 시간이 걸릴 수 있습니다.

검증 단계

- 

사용자 또는 사용자 그룹이 사용자 또는 사용자 그룹의 멤버 관리자 목록에서 제거되었는지 확인합니다.

## User Group: project

project members:

|       |             |          |
|-------|-------------|----------|
| Users | User Groups | Services |
|-------|-------------|----------|

project member managers:

|             |           |
|-------------|-----------|
| User Groups | Users (1) |
|-------------|-----------|

|         |        |     |
|---------|--------|-----|
| Refresh | Delete | Add |
|---------|--------|-----|

|                          |            |
|--------------------------|------------|
| <input type="checkbox"/> | Group name |
| No entries.              |            |

추가 리소스

- 

자세한 내용은 `ipa group-add-member-manager --help` 를 참조하십시오.

## 13장. ANSIBLE 플레이북을 사용하여 사용자 그룹 관리

이 섹션에서는 **Ansible** 플레이북을 사용하여 사용자 그룹 관리를 소개합니다.

사용자 그룹은 공통 권한, 암호 정책 및 기타 특성을 가진 사용자 집합입니다.

**IdM(Identity Management)**의 사용자 그룹에는 다음이 포함될 수 있습니다.

- **IdM 사용자**
- **기타 IdM 사용자 그룹**
- **외부 사용자: IdM 외부에 있는 사용자**

섹션에는 다음 주제가 포함되어 있습니다.

- **IdM의 다양한 그룹 유형**
- **직접 및 간접 그룹 멤버**
- **Ansible 플레이북을 사용하여 IdM 그룹 및 그룹 멤버가 있는지 확인**
- **Ansible 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 있는지 확인**
- **Ansible 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 없는지 확인**

### 13.1. IDM의 다양한 그룹 유형

**IdM**은 다음 유형의 그룹을 지원합니다.

## POSIX 그룹(기본값)

**POSIX** 그룹은 해당 멤버에 대해 **Linux POSIX** 속성을 지원합니다. **Active Directory**와 상호 작용하는 그룹에서는 **POSIX** 속성을 사용할 수 없습니다.

**POSIX** 속성은 사용자를 별도의 엔터티로 식별합니다. 사용자와 관련된 **POSIX** 속성의 예로는 **uidNumber**, **UID**(사용자 번호), **gidNumber**, 그룹 번호(**GID**)가 있습니다.

## 비POSIX 그룹

비**POSIX** 그룹은 **POSIX** 속성을 지원하지 않습니다. 예를 들어 이러한 그룹에는 **GID**가 정의되어 있지 않습니다.

이러한 유형의 그룹의 모든 멤버가 **IdM** 도메인에 속해야 합니다.

## 외부 그룹

외부 그룹을 사용하여 **IdM** 도메인 외부의 **ID** 저장소에 존재하는 그룹 멤버를 다음과 같이 추가합니다.

- 로컬 시스템
- **Active Directory** 도메인
- 디렉터리 서비스

외부 그룹은 **POSIX** 속성을 지원하지 않습니다. 예를 들어 이러한 그룹에는 **GID**가 정의되어 있지 않습니다.

표 13.1. 기본적으로 생성된 사용자 그룹

| 그룹 이름           | 기본 그룹 멤버                                 |
|-----------------|------------------------------------------|
| <b>ipausers</b> | 모든 <b>IdM</b> 사용자                        |
| 관리자             | 기본 <b>admin</b> 사용자를 포함하여 관리자 권한이 있는 사용자 |
| <b>editors</b>  | 이는 더 이상 특별한 권한이 없는 레거시 그룹입니다.            |

| 그룹 이름  | 기본 그룹 멤버                                   |
|--------|--------------------------------------------|
| 신뢰 관리자 | Active Directory 트러스트를 관리할 수 있는 권한이 있는 사용자 |

사용자 그룹에 사용자를 추가하면 사용자와 연결된 권한과 정책이 제공됩니다. 예를 들어 사용자에게 관리 권한을 부여하려면 사용자를 **admins** 그룹에 추가합니다.



#### 주의

**admins** 그룹을 삭제하지 마십시오. 관리자는 IdM에 필요한 사전 정의된 그룹이므로 이 작업으로 인해 특정 명령에 문제가 발생합니다.

또한 IdM은 IdM에 새 사용자를 생성할 때마다 기본적으로 사용자 개인 그룹을 생성합니다. 개인 그룹에 대한 자세한 내용은 개인 그룹 [없이 사용자 추가](#)를 참조하십시오.

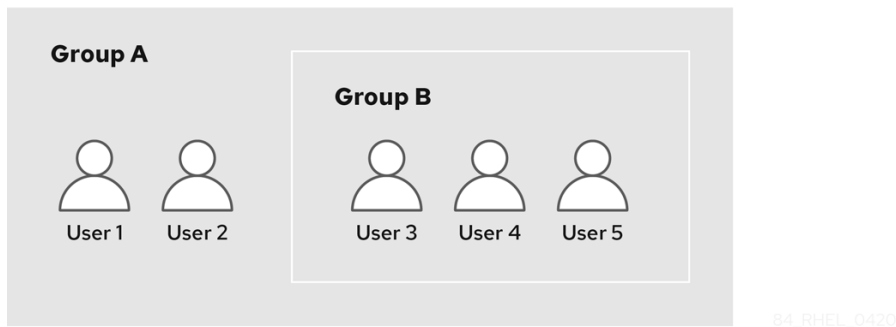
### 13.2. 직접 및 간접 그룹 멤버

IdM의 사용자 그룹 속성은 직접 및 간접 멤버 모두에 적용됩니다. 그룹 B가 A 그룹의 멤버인 경우 B 그룹의 모든 사용자는 A 그룹의 간접 멤버로 간주됩니다.

예를 들어 다음 다이어그램에서: **For example, in the following diagram:**

- 사용자 1과 사용자 2는 A 그룹의 **직접적인 멤버**입니다.
- 사용자 3, 사용자 4 및 사용자 5는 A 그룹의 **간접 멤버**입니다.

그림 13.1. 직접 및 직접 그룹 멤버십



사용자 그룹 **A**에 대한 암호 정책을 설정하는 경우 정책은 사용자 그룹 **B**의 모든 사용자에게도 적용됩니다.

### 13.3. ANSIBLE 플레이북을 사용하여 IDM 그룹 및 그룹 멤버가 있는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 그룹 및 그룹 멤버(사용자 및 사용자 그룹)가 있는지 확인하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.
- **Ansible** 플레이북에서 참조하려는 사용자는 **IdM**에 있습니다. **Ansible**을 사용하여 사용자가 있는지 확인하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 사용자 계정](#) 관리를 참조하십시오.

#### 절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 `ipaserver` 를 이 파일에 정의합니다.

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

2. 필요한 사용자 및 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다.

```
---
```

```

- name: Playbook to handle groups
  hosts: ipaserver
  become: true

  tasks:
    - name: Create group ops with gid 1234
      ipagroup:
        ipaadmin_password: MySecret123
        name: ops
        gidnumber: 1234

    - name: Create group sysops
      ipagroup:
        ipaadmin_password: MySecret123
        name: sysops
        user:
          - idm_user

    - name: Create group appops
      ipagroup:
        ipaadmin_password: MySecret123
        name: appops

    - name: Add group members sysops and appops to group ops
      ipagroup:
        ipaadmin_password: MySecret123
        name: ops
        group:
          - sysops
          - appops

```

3.

플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/add-group-members.yml

```

검증 단계

**ops** 그룹에 **sysops** 및 **app ops** 가 직접 멤버로 포함되어 있는지, **ipa group-show** 명령을 사용하여 **idm\_user** 를 간접 멤버로 포함하고 있는지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

```

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

```

2.

**ops**에 대한 정보를 표시합니다:

```
ipaserver]$ ipa group-show ops
Group name: ops
GID: 1234
Member groups: sysops, appops
Indirect Member users: idm_user
```

**appops** 및 **sysops** 그룹 - 후자의 경우 **idm\_user** 사용자를 포함하여 **IdM**에 있습니다.

추가 리소스

- [/usr/share/doc/ansible-freeipa/README-group.md](#) Markdown 파일을 참조하십시오.

#### 13.4. ANSIBLE 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 있는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 멤버 관리자(사용자 및 사용자 그룹 모두)가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- 멤버 관리자로 추가하는 사용자 또는 그룹의 이름과 관리할 그룹의 이름이 있어야 합니다.

절차

1.

인벤토리 파일(예: **inventory.file**)을 생성하고 **ipaserver**를 이 파일에 정의합니다.

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

2.

필요한 사용자 및 그룹 멤버 관리 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다.

```

---
- name: Playbook to handle membership management
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure user test is present for group_a
      ipagroup:
        ipaadmin_password: MySecret123
        name: group_a
        membermanager_user: test

    - name: Ensure group_admins is present for group_a
      ipagroup:
        ipaadmin_password: MySecret123
        name: group_a
        membermanager_group: group_admins

```

3.

플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/add-member-managers-user-groups.yml

```

검증 단계

**group\_a** 그룹에 멤버 관리자로 **test** 가 포함되어 있고 **group\_admins** 가 **ipa group-show** 명령을 사용하여 **group\_a** 의 멤버 관리자인지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

```

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

```

2.

**managergroup1** 에 대한 정보를 표시합니다.

```

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

```

추가 리소스

- `ipa host-add-member-manager --help` 를 참조하십시오.
- `ipa man` 페이지를 참조하십시오.

### 13.5. ANSIBLE 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 없는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 멤버 관리자(사용자 및 사용자 그룹 모두)가 없는지 확인하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.
- 제거 중인 기존 멤버 관리자 사용자 또는 그룹의 이름과 관리 중인 그룹의 이름이 있어야 합니다.

#### 절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 `ipaserver` 를 이 파일에 정의합니다.

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

2. 필요한 사용자 및 그룹 멤버 관리 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다.

```
---
- name: Playbook to handle membership management
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure member manager user and group members are absent for group_a
      ipagroup:
        ipadmin_password: MySecret123
        name: group_a
        membermanager_user: test
```

```
membermanager_group: group_admins
action: member
state: absent
```

3.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-member-managers-are-absent.yml
```

## 검증 단계

**ipa group-show** 명령을 사용하여 **group\_a** 그룹에 **test** 가 멤버 관리자로, **group\_admins** 를 **group\_a** 의 멤버 관리자로 포함하고 있지 않은지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

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

2.

**group\_a**에 대한 정보를 표시합니다.

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

## 추가 리소스

- **ipa host-remove-member-manager --help** 를 참조하십시오.
- **ipa man** 페이지를 참조하십시오.

## 14장. IDM CLI를 사용하여 그룹 멤버십 자동화

자동 그룹 멤버십을 사용하면 특성을 기반으로 사용자와 호스트를 그룹에 자동으로 할당할 수 있습니다. 예를 들면 다음을 수행할 수 있습니다.

- 직원의 사용자 항목을 직원의 관리자, 위치 또는 기타 속성에 따라 그룹으로 나눕니다.
- 클래스, 위치 또는 기타 특성을 기반으로 호스트를 나눕니다.
- 단일 글로벌 그룹에 모든 사용자 또는 모든 호스트를 추가합니다.

이 장에서는 다음 주제를 다룹니다.

- [자동 그룹 멤버십의 이점](#)
- [자동 멤버 규칙](#)
- [IdM CLI를 사용하여 automember 규칙 추가](#)
- [IdM CLI를 사용하여 automember 규칙에 조건 추가](#)
- [IdM CLI를 사용하여 기존 automember 규칙 보기](#)
- [IdM CLI를 사용하여 automember 규칙 삭제](#)
- [IdM CLI를 사용하여 automember 규칙에서 조건 제거](#)
- [IdM CLI를 사용하여 기존 항목에 자동 멤버십 규칙 적용](#)

- **IdM CLI를 사용하여 기본 `automember` 그룹 구성**

### 14.1. 자동 그룹 멤버십의 이점

사용자에 대해 자동 멤버십을 사용하면 다음을 수행할 수 있습니다.

- 그룹 멤버십을 수동으로 관리하는 오버헤드 감소  
  
더 이상 모든 사용자와 호스트를 수동으로 그룹에 할당할 필요가 없습니다.
- 사용자 및 호스트 관리의 일관성 향상  
  
사용자와 호스트는 엄격하게 정의되고 자동으로 평가된 기준에 따라 그룹에 할당됩니다.
- 그룹 기반 설정 관리 단순화  
  
그룹에 대해 다양한 설정이 정의된 다음, 개별 그룹 멤버(예: `sudo` 규칙, 자동 마운트 또는 액세스 제어)에 적용됩니다. 그룹에 사용자와 호스트를 자동으로 추가하면 이러한 설정을 보다 쉽게 관리할 수 있습니다.

### 14.2. 자동 멤버 규칙

자동 그룹 멤버십을 구성할 때 관리자는 **automember** 규칙을 정의합니다. **automember** 규칙은 특정 사용자 또는 호스트 대상 그룹에 적용됩니다. 한 번에 둘 이상의 그룹에 적용할 수 없습니다.

규칙을 생성한 후 관리자는 여기에 조건을 추가합니다. 이는 대상 그룹에서 포함하거나 제외되는 사용자 또는 호스트를 지정합니다.

- 포함된 조건  
  
사용자 또는 호스트 항목이 포함 조건을 충족하면 대상 그룹에 포함됩니다.
- 독점 조건

사용자 또는 호스트 항목이 배타적인 조건을 충족하면 대상 그룹에 포함되지 않습니다.

조건은 **Perl** 호환 정규식(**PCRE**) 형식의 정규식으로 지정됩니다. **PCRE**에 대한 자세한 내용은 **PC resyntax(3)** 매뉴얼 페이지를 참조하십시오.



참고

**IdM**은 포함 조건 이전에 배타적 조건을 평가합니다. 충돌이 발생하는 경우 예외 조건이 포괄적 조건보다 우선합니다.

**automember** 규칙은 향후 생성된 모든 항목에 적용됩니다. 이러한 항목은 지정된 대상 그룹에 자동으로 추가됩니다. 항목이 여러 **automember** 규칙에 지정된 조건을 충족하면 해당 그룹에 모두 추가됩니다.

기존 항목은 새 규칙의 영향을 받지 않습니다. 기존 항목을 변경하려면 **IdM CLI**를 사용하여 기존 항목에 자동 멤버십 규칙 적용을 참조하십시오.

### 14.3. IDM CLI를 사용하여 AUTOMEMBER 규칙 추가

이 섹션에서는 **IdM CLI**를 사용하여 **automember** 규칙 추가를 설명합니다. **automember** 규칙에 대한 자세한 내용은 **Automember rules** 을 참조하십시오.

**automember** 규칙을 추가한 후 **Adding a condition to an automember 규칙**에 설명된 절차를 사용하여 조건을 추가할 수 있습니다.



참고

기존 항목은 새 규칙의 영향을 받지 않습니다. 기존 항목을 변경하려면 **IdM CLI**를 사용하여 기존 항목에 자동 멤버십 규칙 적용을 참조하십시오.

#### 사전 요구 사항

- 

관리자로 로그인해야 합니다. 자세한 내용은 **kinit**를 사용하여 **IdM**에 수동으로 로그인하는 것을 참조하십시오.

- 새 규칙의 대상 그룹이 **IdM**에 있어야 합니다.

#### 절차

1. **ipa automember-add** 명령을 입력하여 **automember** 규칙을 추가합니다.
  2. 메시지가 표시되면 다음을 지정합니다.
- 자동 멤버 규칙. 대상 그룹 이름입니다.
  - 그룹화 유형. 이는 규칙이 사용자 그룹 또는 호스트 그룹을 대상으로 하는지 여부를 지정합니다. 사용자 그룹을 대상으로 지정하려면 그룹을 입력합니다. 호스트 그룹을 대상으로 지정하려면 **hostgroup** 을 입력합니다.

예를 들어 **user\_group** 이라는 사용자 그룹에 대한 **automember** 규칙을 추가하려면 다음을 수행합니다.

```
$ ipa automember-add
Automember Rule: user_group
Grouping Type: group
-----
Added automember rule "user_group"
-----
Automember Rule: user_group
```

#### 검증 단계

- **IdM CLI**를 사용하여 기존 **automember** 규칙 보기를 사용하여 **IdM**에 기존 자동 멤버십 규칙 및 조건을 표시할 수 있습니다.

### 14.4. IDM CLI를 사용하여 AUTOMEMBER 규칙에 조건 추가

이 섹션에서는 **IdM CLI**를 사용하여 **automember** 규칙에 조건을 추가하는 방법을 설명합니다. **automember** 규칙에 대한 자세한 내용은 [Automember rules](#) 을 참조하십시오.

#### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit](#)를 사용하여 **IdM**에 수동으로 로그인하는 것

을 참조하십시오.

- **IdM에 대상 규칙이 있어야 합니다.** 자세한 내용은 **IdM CLI를 사용하여 자동 멤버십 규칙 추가**를 참조하십시오.

## 절차

1. **ipa automember-add-condition** 명령을 사용하여 하나 이상의 포함 또는 전용 조건을 정의합니다.
2. 메시지가 표시되면 다음을 지정합니다.
  - 자동 멤버 규칙. 대상 규칙 이름입니다. 자세한 내용은 **Automember 규칙**을 참조하십시오.
  - 특성 키. 이는 필터를 적용할 항목 속성을 지정합니다. 예를 들어 사용자를 위한 **uid**가 있습니다.
  - 그룹화 유형. 이는 규칙이 사용자 그룹 또는 호스트 그룹을 대상으로 하는지 여부를 지정합니다. 사용자 그룹을 대상으로 지정하려면 그룹을 입력합니다. 호스트 그룹을 대상으로 지정하려면 **hostgroup**을 입력합니다.
  - 정규 표현식 및 **Exclusive regex**. 하나 이상의 조건을 정규 표현식으로 지정합니다. 하나의 조건만 지정하려면 다른 조건을 입력하라는 메시지가 표시되면 **Enter**를 누릅니다.

예를 들어 다음 조건은 사용자 로그인 속성(**uid**)에서 모든 값(.)을 사용하여 모든 사용자를 대상으로 합니다.

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

다른 예로, **automembership** 규칙을 사용하여 **Active Directory (AD)**에서 동기화 된 모든 **Windows** 사용자를 대상으로 할 수 있습니다. 이를 위해 모든 **AD** 사용자가 공유하는 **objectClass** 속성에서 **ntUser** 를 사용하는 모든 사용자를 대상으로 하는 조건을 생성합니다.

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

#### 검증 단계

- **IdM CLI**를 사용하여 기존 **automember** 규칙 보기를 사용하여 **IdM**에 기존 자동 멤버십 규칙 및 조건을 표시할 수 있습니다.

### 14.5. IDM CLI를 사용하여 기존 AUTOMEMBER 규칙 보기

이 섹션에서는 **IdM CLI**를 사용하여 기존 **automember** 규칙을 보는 방법을 설명합니다.

#### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 **kinit**를 사용하여 **IdM**에 수동으로 로그인하는 것을 참조하십시오.

#### 절차

1. **ipa automember-find** 명령을 입력합니다.
2. 메시지가 표시되면 그룹화 유형을 지정합니다.

- 사용자 그룹을 대상으로 지정하려면 그룹을 입력합니다.
- 호스트 그룹을 대상으로 지정하려면 **hostgroup** 을 입력합니다.

예를 들어 다음과 같습니다.

```
$ ipa automember-find
Grouping Type: group
-----
1 rules matched
-----
Automember Rule: user_group
Inclusive Regex: uid=.*
-----
Number of entries returned 1
-----
```

#### 14.6. IDM CLI를 사용하여 AUTOMEMBER 규칙 삭제

이 섹션에서는 **IdM CLI**를 사용하여 **automember** 규칙을 삭제하는 방법을 설명합니다.

**automember** 규칙을 삭제하면 규칙과 관련된 모든 조건도 삭제됩니다. 규칙에서 특정 조건만 제거하려면 **IdM CLI를 사용하여 automember 규칙에서 조건 제거**를 참조하십시오.

##### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#)을 참조하십시오.

##### 절차

1. **ipa automember-del** 명령을 입력합니다.
2. 메시지가 표시되면 다음을 지정합니다.
  - 자동 멤버 규칙. 삭제하려는 규칙입니다.
  -

그룹화 규칙. 삭제하려는 규칙이 사용자 그룹 또는 호스트 그룹에 대한지 여부를 지정합니다. 그룹 또는 호스트 그룹을 입력합니다.

## 14.7. IDM CLI를 사용하여 AUTOMEMBER 규칙에서 조건 제거

이 섹션에서는 **automember** 규칙에서 특정 조건을 제거하는 방법을 설명합니다.

### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#)을 참조하십시오.

### 절차

1. **ipa automember-remove-condition** 명령을 입력합니다.
2. 메시지가 표시되면 다음을 지정합니다.
  - 자동 멤버 규칙. 조건을 제거할 규칙의 이름입니다.
  - 특성 키. 대상 항목 특성입니다. 예를 들어 사용자를 위한 **uid**가 있습니다.
  - 그룹화 유형. 삭제하려는 조건이 사용자 그룹 또는 호스트 그룹에 대한 것인지 여부를 지정합니다. 그룹 또는 호스트 그룹을 입력합니다.
  - 정규 표현식 및 **Exclusive regex**. 이러한 조건은 제거할 조건을 지정합니다. 하나의 조건만 지정하려면 다른 조건을 입력하라는 메시지가 표시되면 **Enter**를 누릅니다.

예를 들어 다음과 같습니다.

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

#### 14.8. IDM CLI를 사용하여 기존 항목에 자동 멤버십 규칙 적용

자동 멤버십 규칙은 규칙을 추가한 후 사용자 및 호스트 항목에 자동으로 적용됩니다. 규칙이 추가되기 전에 존재하는 항목에 소급적 적용되지 않습니다.

이전에 추가된 항목에 **automember** 규칙을 적용하려면 자동 멤버십을 수동으로 다시 빌드해야 합니다. 자동 멤버십을 다시 작성하여 기존의 모든 자동 멤버십 규칙을 다시 평가하여 모든 사용자 또는 호스트 항목 또는 특정 항목에 적용합니다.



##### 참고

자동 멤버십을 다시 작성해도 그룹에서 사용자 또는 호스트 항목을 제거하지는 않습니다. 항목이 더 이상 그룹의 포함 조건과 일치하지 않는 경우에도 마찬가지입니다. 수동으로 제거하려면 **CLI를 사용하여 IdM CLI를 사용하여 사용자 그룹에서 멤버 제거 또는 IdM 호스트 그룹 멤버 제거**를 참조하십시오.

##### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 **kinit를 사용하여 IdM에 수동으로 로그인하는 것**을 참조하십시오.

##### 절차

- 자동 멤버십을 다시 빌드하려면 **ipa automember-rebuild** 명령을 입력합니다. 다음 옵션을 사용하여 대상 항목을 지정합니다.
  - 모든 사용자에게 대해 자동 멤버십을 다시 빌드하려면 **--type=group** 옵션을 사용합니다.

```
$ ipa automember-rebuild --type=group
```

```
-----
Automember rebuild task finished. Processed (9) entries.
-----
```

- 모든 호스트에 대한 자동 멤버십을 다시 빌드하려면 **--type=hostgroup** 옵션을 사용합

니다.

- 지정된 사용자 또는 사용자의 자동 멤버십을 다시 작성하려면 `--users=target_user` 옵션을 사용합니다.

```
$ ipa automember-rebuild --users=target_user1 --users=target_user2
```

```
-----
Automember rebuild task finished. Processed (2) entries.
-----
```

- 지정된 호스트 또는 호스트의 자동 멤버십을 다시 빌드하려면 `--hosts=client.idm.example.com` 옵션을 사용합니다.

## 14.9. IDM CLI를 사용하여 기본 AUTOMEMBER 그룹 구성

기본 **automember** 그룹을 구성하면 **automember** 규칙과 일치하지 않는 새 사용자 또는 호스트 항목이 자동으로 이 기본 그룹에 추가됩니다.

### 사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것](#)을 참조하십시오.
- IdM에 기본값으로 설정할 대상 그룹이 있습니다.

### 절차

1. `ipa automember-default-group-set` 명령을 입력하여 기본 **automember** 그룹을 구성합니다.
2. 메시지가 표시되면 다음을 지정합니다.
  - 대상 그룹 이름을 지정하는 기본 (fallback) 그룹 .
  - 그룹화 유형: 대상이 사용자 그룹인지 아니면 호스트 그룹인지 여부를 지정합니다. **Grouping Type, which specifies whether the target is a user group or a host group.** 사

용자 그룹을 대상으로 지정하려면 그룹을 입력합니다. 호스트 그룹을 대상으로 지정하려면 **hostgroup** 을 입력합니다.

예를 들어 다음과 같습니다.

```
$ 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
```



참고

현재 기본 **automember** 그룹을 제거하려면 **ipa automember-default-group-remove** 명령을 입력합니다.

검증 단계

•

그룹이 올바르게 설정되었는지 확인하려면 **ipa automember-default-group-show** 명령을 입력합니다. 명령은 현재 기본 **automember** 그룹을 표시합니다. 예를 들어 다음과 같습니다.

```
$ ipa automember-default-group-show
Grouping Type: group
Default (fallback) Group:
cn=default_user_group,cn=groups,cn=accounts,dc=example,dc=com
```

## 15장. IDM 웹 UI를 사용하여 그룹 멤버십 자동화

자동 그룹 멤버십을 사용하면 특성을 기반으로 사용자와 호스트를 그룹에 자동으로 할당할 수 있습니다. 예를 들면 다음을 수행할 수 있습니다.

- 직원의 사용자 항목을 직원의 관리자, 위치 또는 기타 속성에 따라 그룹으로 나눕니다.
- 클래스, 위치 또는 기타 특성을 기반으로 호스트를 나눕니다.
- 단일 글로벌 그룹에 모든 사용자 또는 모든 호스트를 추가합니다.

이 장에서는 다음 주제를 다룹니다.

- [자동 그룹 멤버십의 이점](#)
- [자동 멤버 규칙](#)
- [IdM 웹 UI를 사용하여 automember 규칙 추가](#)
- [IdM 웹 UI를 사용하여 자동 멤버 규칙에 조건 추가](#)
- [IdM 웹 UI를 사용하여 기존 automember 규칙 및 조건 보기](#)
- [IdM 웹 UI를 사용하여 automember 규칙 삭제](#)
- [IdM 웹 UI를 사용하여 자동 멤버 규칙에서 조건 제거](#)
- [IdM 웹 UI를 사용하여 기존 항목에 자동 멤버십 규칙 적용](#)

- [IdM 웹 UI를 사용하여 기본 사용자 그룹 구성](#)
- [IdM 웹 UI를 사용하여 기본 호스트 그룹 구성](#)

### 15.1. 자동 그룹 멤버십의 이점

사용자에 대해 자동 멤버십을 사용하면 다음을 수행할 수 있습니다.

- 그룹 멤버십을 수동으로 관리하는 오버헤드 감소  
  
더 이상 모든 사용자와 호스트를 수동으로 그룹에 할당할 필요가 없습니다.
- 사용자 및 호스트 관리의 일관성 향상  
  
사용자와 호스트는 엄격하게 정의되고 자동으로 평가된 기준에 따라 그룹에 할당됩니다.
- 그룹 기반 설정 관리 단순화  
  
그룹에 대해 다양한 설정이 정의된 다음, 개별 그룹 멤버(예: **sudo** 규칙, 자동 마운트 또는 액세스 제어)에 적용됩니다. 그룹에 사용자와 호스트를 자동으로 추가하면 이러한 설정을 보다 쉽게 관리할 수 있습니다.

### 15.2. 자동 멤버 규칙

자동 그룹 멤버십을 구성할 때 관리자는 **automember** 규칙을 정의합니다. **automember** 규칙은 특정 사용자 또는 호스트 대상 그룹에 적용됩니다. 한 번에 둘 이상의 그룹에 적용할 수 없습니다.

규칙을 생성한 후 관리자는 여기에 조건을 추가합니다. 이는 대상 그룹에서 포함하거나 제외되는 사용자 또는 호스트를 지정합니다.

- 포함된 조건  
  
사용자 또는 호스트 항목이 포함 조건을 충족하면 대상 그룹에 포함됩니다.



### 독점 조건

사용자 또는 호스트 항목이 배타적인 조건을 충족하면 대상 그룹에 포함되지 않습니다.

조건은 **Perl** 호환 정규식(**PCRE**) 형식의 정규식으로 지정됩니다. **PCRE**에 대한 자세한 내용은 **PC resyntax(3)** 매뉴얼 페이지를 참조하십시오.



### 참고

**IdM**은 포함 조건 이전에 배타적 조건을 평가합니다. 충돌이 발생하는 경우 예외 조건이 포괄적 조건보다 우선합니다.

**automember** 규칙은 향후 생성된 모든 항목에 적용됩니다. 이러한 항목은 지정된 대상 그룹에 자동으로 추가됩니다. 항목이 여러 **automember** 규칙에 지정된 조건을 충족하면 해당 그룹에 모두 추가됩니다.

기존 항목은 새 규칙의 영향을 받지 않습니다. 기존 항목을 변경하려면 **IdM 웹 UI를 사용하여 기존 항목에 자동 멤버십 규칙 적용**을 참조하십시오.

## 15.3. IdM 웹 UI를 사용하여 AUTOMEMBER 규칙 추가

이 섹션에서는 **IdM 웹 UI**를 사용하여 자동 멤버십 규칙을 추가하는 방법을 설명합니다. **automember** 규칙에 대한 자세한 내용은 **Automember rules**을 참조하십시오.



### 참고

기존 항목은 새 규칙의 영향을 받지 않습니다. 기존 항목을 변경하려면 **IdM 웹 UI를 사용하여 기존 항목에 자동 멤버십 규칙 적용**을 참조하십시오.

### 사전 요구 사항



**IdM 웹 UI**에 로그인되어 있습니다.

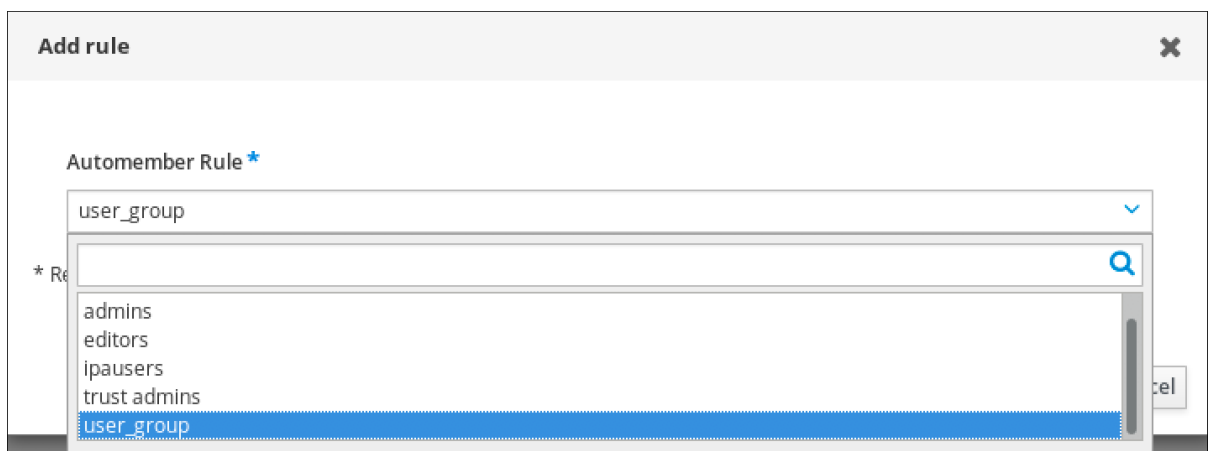


**admins** 그룹의 멤버여야 합니다.

- **IdM에 새 규칙의 대상 그룹이 있습니다.**

#### 절차

1. **Identity → Automember** 를 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택합니다.
2. **추가**를 클릭합니다.
3. **Automember** 규칙 필드에서 규칙이 적용되는 그룹을 선택합니다. 대상 그룹 이름입니다.



4. **Add** 를 클릭하여 확인합니다.
5. 선택 사항: **IdM 웹 UI를 사용하여 automember** 규칙에 조건 추가에 설명된 절차를 사용하여 **새 규칙**에 조건을 추가할 수 있습니다.

### 15.4. IDM 웹 UI를 사용하여 자동 멤버 규칙에 조건 추가

이 섹션에서는 **IdM 웹 UI**를 사용하여 **automember** 규칙에 조건을 추가하는 방법을 설명합니다. **automember** 규칙에 대한 자세한 내용은 [Automember rules](#) 을 참조하십시오.

#### 사전 요구 사항

- **IdM 웹 UI에 로그인되어 있습니다.**

- **admins** 그룹의 멤버여야 합니다.
- **IdM**에 대상 규칙이 있습니다.

#### 절차

1. **Identity** → **Automember** 를 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택합니다.
2. 조건을 추가할 규칙을 클릭합니다.
3. 포함 또는 포함 섹션에서 추가를 클릭합니다. **In the inclusive or Exclusive sections, click Add.**

User group rule: user\_group

### General

**Automember Rule**

user\_group

**Description**

### Inclusive

| <input type="checkbox"/> | Attribute | Expression | Delete <span style="float: right; border: 2px solid red; padding: 2px;">+ Add</span> |
|--------------------------|-----------|------------|--------------------------------------------------------------------------------------|
| <input type="checkbox"/> | uid       | .*         |                                                                                      |

### Exclusive

| <input type="checkbox"/> | Attribute | Expression | Delete <span style="float: right; border: 2px solid red; padding: 2px;">+ Add</span> |
|--------------------------|-----------|------------|--------------------------------------------------------------------------------------|
| <input type="checkbox"/> |           |            |                                                                                      |

4. **Attribute** 필드에서 필요한 속성을 선택합니다(예: *uid*).
5. 정규식 필드에서 정규식을 정의합니다. **In the Expression field, define a regular**

expression.

6. 추가를 클릭합니다.

예를 들어 다음 조건은 사용자 ID(uid) 특성의 모든 값(.)을 사용하여 모든 사용자를 대상으로 합니다.

### 15.5. IDM 웹 UI를 사용하여 기존 AUTOMEMBER 규칙 및 조건 보기

이 섹션에서는 IdM 웹 UI를 사용하여 기존 automember 규칙 및 조건을 보는 방법을 설명합니다.

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- admins 그룹의 멤버여야 합니다.

절차

1. **Identity** → **Automember** 를 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택하여 해당 자동 멤버십 규칙을 확인합니다.
2. 선택 사항: 규칙을 클릭하여 결정적 또는 결정적 섹션에서 해당 규칙에 대한 조건을 확인합니다.

User group rule: user\_group

General

Automember Rule

user\_group

Description

**Inclusive**

| <input type="checkbox"/> | Attribute | Expression | Delete + Add |
|--------------------------|-----------|------------|--------------|
| <input type="checkbox"/> | uid       | ,*         |              |

**Exclusive**

| <input type="checkbox"/> | Attribute | Expression | Delete + Add |
|--------------------------|-----------|------------|--------------|
| <input type="checkbox"/> |           |            |              |

## 15.6. IDM 웹 UI를 사용하여 AUTOMEMBER 규칙 삭제

이 섹션에서는 IdM 웹 UI를 사용하여 automember 규칙을 삭제하는 방법을 설명합니다.

automember 규칙을 삭제하면 규칙과 관련된 모든 조건도 삭제됩니다. 규칙에서 특정 조건만 제거하려면 [IdM 웹 UI를 사용하여 자동 멤버십 규칙에서 조건 제거](#)를 참조하십시오.

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- admins 그룹의 멤버여야 합니다.

절차

1. **Identity** → **Automember** 를 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택하여 해당 자동 멤버십 규칙을 확인합니다.

2. 제거할 규칙 옆에 있는 확인란을 선택합니다.
3. 삭제를 클릭합니다.

User group rules

Search

Default user group

| <input type="checkbox"/>            | Automember Rule | Description |
|-------------------------------------|-----------------|-------------|
| <input checked="" type="checkbox"/> | user_group      |             |

1 rules matched

4. **Delete** 를 클릭하여 확인합니다.

## 15.7. IDM 웹 UI를 사용하여 자동 멤버 규칙에서 조건 제거

이 섹션에서는 **IdM 웹 UI**를 사용하여 자동 멤버십 규칙에서 특정 조건을 제거하는 방법을 설명합니다.

### 사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.

### 절차

1. **Identity** → **Automember** 를 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택하여 해당 자동 멤버십 규칙을 확인합니다.
2. 규칙을 클릭하여 결정적 또는 결정적 섹션에서 해당 규칙에 대한 조건을 확인합니다.
3. 제거할 조건 옆에 있는 확인란을 선택합니다.

4.

삭제를 클릭합니다.

User group rule: user\_group

Refresh Revert Save

General

Automember Rule

user\_group

Description

Inclusive

| <input type="checkbox"/>            | Attribute | Expression | Delete + Add |
|-------------------------------------|-----------|------------|--------------|
| <input checked="" type="checkbox"/> | uid       | .*         | Delete + Add |

Exclusive

| <input type="checkbox"/> | Attribute | Expression | Delete + Add |
|--------------------------|-----------|------------|--------------|
|--------------------------|-----------|------------|--------------|

5.

Delete 를 클릭하여 확인합니다.

## 15.8. IDM 웹 UI를 사용하여 기존 항목에 자동 멤버십 규칙 적용

자동 멤버십 규칙은 규칙을 추가한 후 사용자 및 호스트 항목에 자동으로 적용됩니다. 규칙이 추가되기 전에 존재하는 항목에 소급적 적용되지 않습니다.

이전에 추가된 항목에 **automember** 규칙을 적용하려면 자동 멤버십을 수동으로 다시 빌드해야 합니다. 자동 멤버십을 다시 작성하여 기존의 모든 자동 멤버십 규칙을 다시 평가하여 모든 사용자 또는 호스트 항목 또는 특정 항목에 적용합니다.



### 참고

자동 멤버십을 다시 작성해도 그룹에서 사용자 또는 호스트 항목을 제거하지는 않습니다. 항목이 더 이상 그룹의 포함 조건과 일치하지 않는 경우에도 마찬가지입니다. 수동으로 제거하려면 **IdM 웹 UI** 를 사용하여 사용자 그룹에서 멤버 제거 또는 **IdM 웹 UI** 에서 호스트 그룹 멤버 제거를 참조하십시오.

### 15.8.1. 모든 사용자 또는 호스트에 대해 자동 멤버십 다시 작성

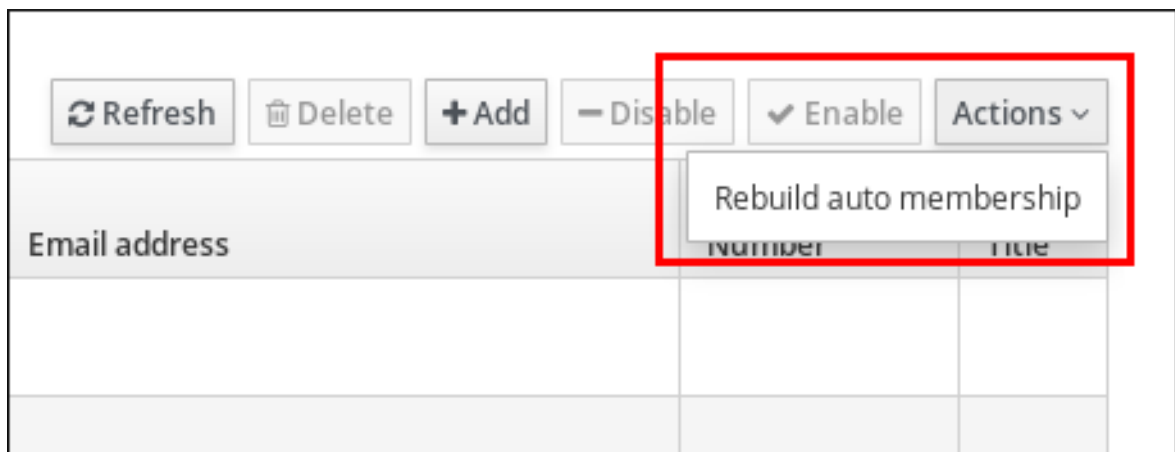
이 섹션에서는 모든 사용자 또는 호스트 항목에 대해 자동 멤버십을 다시 빌드하는 방법을 설명합니다.

사전 요구 사항

- **IdM 웹 UI에 로그인되어 있습니다.**
- **admins 그룹의 멤버여야 합니다.**

절차

1. **ID → 사용자 또는 호스트 를 선택합니다.**
2. **작업 → 자동 멤버십을 다시 빌드합니다.**



### 15.8.2. 단일 사용자 또는 호스트에 대해서만 자동 멤버십 다시 작성

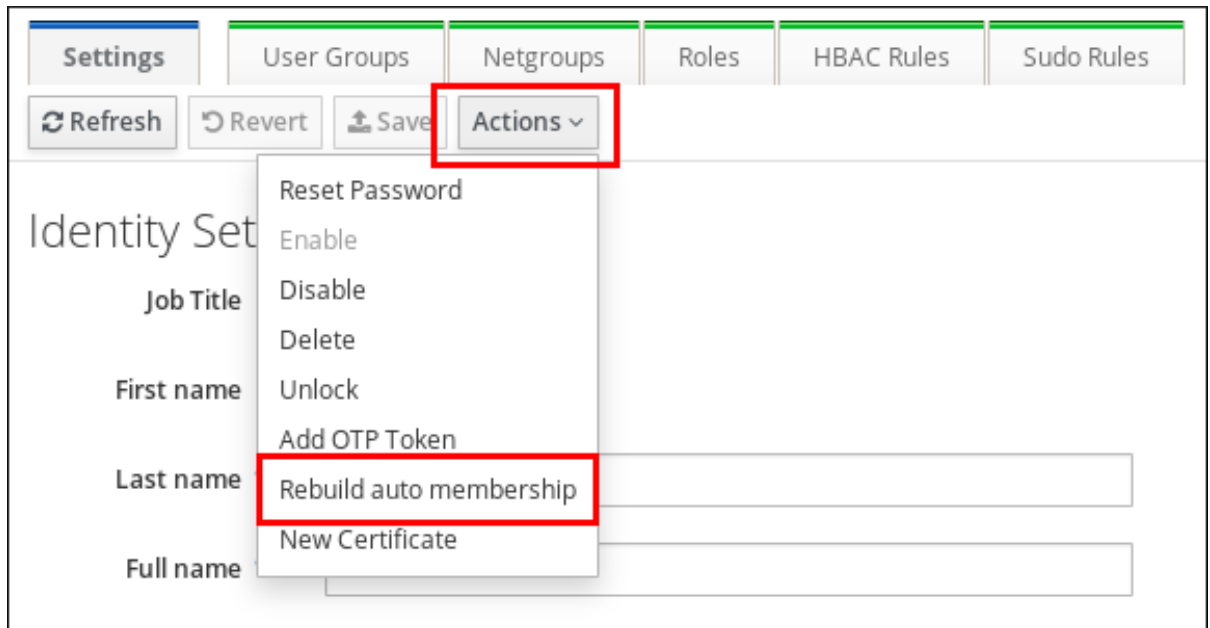
이 섹션에서는 특정 사용자 또는 호스트 항목에 대해 자동 멤버십을 다시 빌드하는 방법을 설명합니다.

사전 요구 사항

- **IdM 웹 UI에 로그인되어 있습니다.**
- **admins 그룹의 멤버여야 합니다.**

## 절차

1. **ID** → 사용자 또는 호스트 를 선택합니다.
2. 필요한 사용자 또는 호스트 이름을 클릭합니다.
3. 작업 → 자동 멤버십을 다시 빌드합니다.



## 15.9. IDM 웹 UI를 사용하여 기본 사용자 그룹 구성

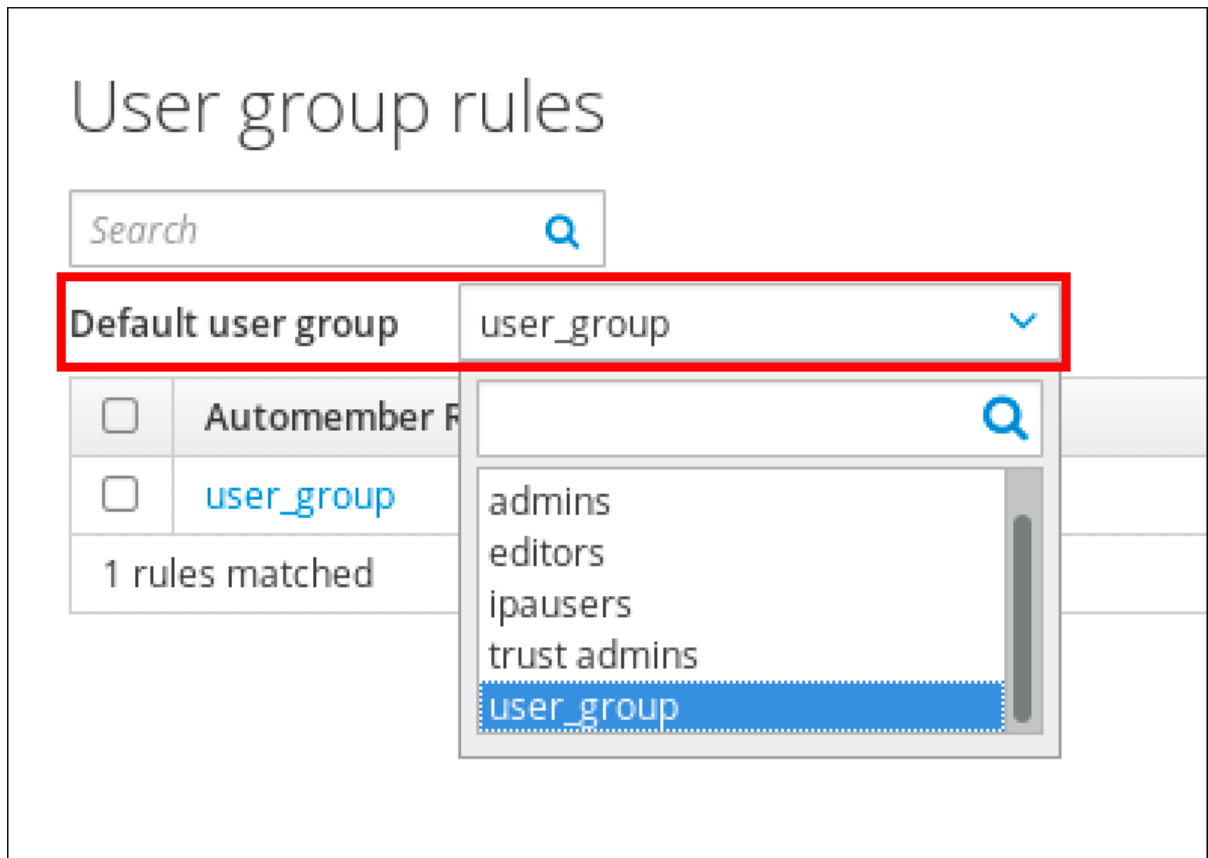
기본 사용자 그룹을 구성하면 **automember** 규칙과 일치하지 않는 새 사용자 항목이 이 기본 그룹에 자동으로 추가됩니다.

## 사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.
- **IdM**에 기본값으로 설정할 대상 사용자 그룹이 있습니다.

## 절차

1. **Identity** → **Automember** 를 클릭하고 사용자 그룹 규칙을 선택합니다.
2. **Default** 사용자 그룹 필드에서 기본 사용자 그룹으로 설정할 그룹을 선택합니다.



### 15.10. IDM 웹 UI를 사용하여 기본 호스트 그룹 구성

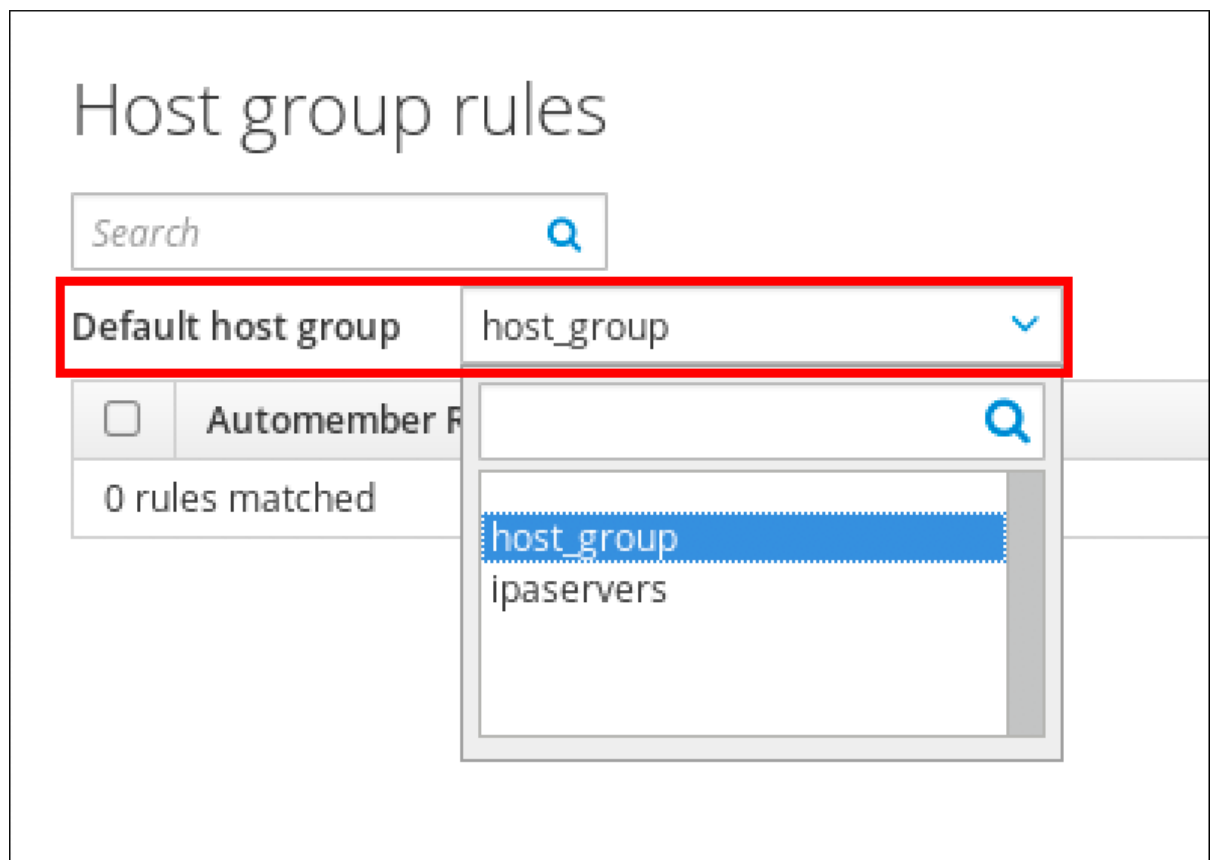
기본 호스트 그룹을 구성하면 **automember** 규칙과 일치하지 않는 새 호스트 항목이 이 기본 그룹에 자동으로 추가됩니다.

#### 사전 요구 사항

- IDM 웹 UI에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.
- IDM에 기본값으로 설정할 대상 호스트 그룹이 있습니다.

#### 절차

1. **Identity** → **Automember** 를 클릭하고 호스트 그룹 규칙을 선택합니다.
2. **Default** 호스트 그룹 필드에서 기본 호스트 그룹으로 설정할 그룹을 선택합니다.



## 16장. ANSIBLE을 사용하여 IDM의 그룹 멤버십 자동화

자동 그룹 멤버십을 사용하여 속성에 따라 사용자 및 호스트 사용자 그룹 및 호스트 그룹을 자동으로 할당할 수 있습니다. 예를 들면 다음을 수행할 수 있습니다.

- 직원의 사용자 항목을 직원의 관리자, 위치, 위치 또는 기타 속성에 따라 그룹으로 나눕니다. 명령줄에서 **ipa user-add --help** 를 입력하여 모든 속성을 나열할 수 있습니다.
- 클래스, 위치 또는 기타 속성에 따라 호스트를 그룹으로 나눕니다. 명령줄에서 **ipa host-add --help** 를 입력하여 모든 속성을 나열할 수 있습니다.
- 단일 글로벌 그룹에 모든 사용자 또는 모든 호스트를 추가합니다.

Red Hat Ansible Engine을 사용하여 IdM(Identity Management)에서 자동 그룹 멤버십 관리를 자동화할 수 있습니다.

이 섹션에서는 다음 주제를 다룹니다.

- **IdM 관리를 위한 Ansible 제어 노드 준비**
- **Ansible을 사용하여 IdM 사용자 그룹에 대한 automember 규칙이 있는지 확인합니다.**
- **Ansible을 사용하여 IdM 사용자 그룹 automember 규칙에 조건이 있는지 확인합니다.**
- **Ansible을 사용하여 IdM 사용자 그룹 automember 규칙에 조건이 없는지 확인합니다.**
- **Ansible을 사용하여 IdM 그룹의 automember 규칙이 없는지 확인합니다.**
- **Ansible을 사용하여 IdM 호스트 그룹 automember 규칙에 조건이 있는지 확인합니다.**

### 16.1. IDM 관리를 위한 ANSIBLE 제어 노드 준비

Red Hat Ansible Engine으로 작업할 때 IdM(Identity Management)을 관리하는 시스템 관리자는 다음을 수행하는 것이 좋습니다.

- 홈 디렉터리에서 **Ansible** 플레이북 전용 하위 디렉터리를 만듭니다(예: `~/MyPlaybooks` ).
- `/usr/share/doc/ansible-freeipa/*` 및 `/usr/share/doc/rhel-system-roles/*` 디렉터리 및 하위 디렉터리에서 샘플 **Ansible** 플레이북을 `~/MyPlaybooks` 디렉터리에 복사 및 조정합니다.
- `~/MyPlaybooks` 디렉터리에 인벤토리 파일을 포함합니다.

이 방법을 사용하면 모든 플레이북을 한 곳에서 찾을 수 있으며 **root** 권한을 호출하지 않고도 플레이북을 실행할 수 있습니다.



#### 참고

**ipaserver, ipareplica, ipaclient** 및 **ipabackup ansible-freeipa** 역할을 실행하려면 관리형 노드에서 **root** 권한만 있으면 됩니다. 이러한 역할을 수행하려면 디렉터리 및 **dnf** 소프트웨어 패키지 관리자에 대한 권한 있는 액세스 권한이 필요합니다.

이 섹션에서는 `~/MyPlaybooks` 디렉터리를 생성하고 이를 사용하여 **Ansible** 플레이북을 저장하고 실행하는 방법을 설명합니다.

#### 사전 요구 사항

- 관리 노드, `server.idm.example.com` 및 `replica.idm.example.com` 에 IdM 서버를 설치했습니다.
- 제어 노드에서 직접 관리 노드 `server.idm.example.com` 및 `replica.idm.example.com` 에 로그인할 수 있도록 **DNS** 및 네트워킹을 구성했습니다.
- IdM 관리자 암호를 알고 있습니다.

#### 절차

1.

홈 디렉터리에 **Ansible** 구성 및 플레이북의 디렉터리를 생성합니다.

```
$ mkdir ~/MyPlaybooks/
```

2.

~/MyPlaybooks/ 디렉터리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3.

다음 콘텐츠를 사용하여 ~/MyPlaybooks/ansible.cfg 파일을 생성합니다.

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

[privilege_escalation]
become=True
```

4.

다음 콘텐츠를 사용하여 ~/MyPlaybooks/inventory 파일을 만듭니다.

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

이 구성은 해당 위치에 있는 호스트에 대해 **eu** 및 **us**의 두 개의 호스트 그룹을 정의합니다. 또한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트가 포함된 **ipaserver** 호스트 그룹을 정의합니다.

5.

[선택 사항] **SSH** 공개 및 개인 키를 생성합니다. 테스트 환경에서 액세스를 단순화하려면 개인 키에 암호를 설정하지 마십시오.

```
$ ssh-keygen
```

6.

**SSH** 공개 키를 각 관리 노드의 **IdM** 관리자 계정에 복사합니다.

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

이러한 명령을 입력하면 **IdM** 관리자 암호를 입력해야 합니다.

추가 리소스

- [Ansible 플레이북을 사용하여 ID 관리 서버 설치.](#)
- [인벤토리를 빌드하는 방법.](#)

**16.2. ANSIBLE**을 사용하여 **IDM** 사용자 그룹에 대한 **AUTOMEMBER** 규칙이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM**(Identity Management) 그룹에 대한 자동 멤버십 규칙이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 **testing\_group** 사용자 그룹에 대해 **automember** 규칙이 있는지 확인합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **IdM**에 **testing\_group** 사용자 그룹이 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - **~/MyPlaybooks/** 디렉터리에서 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2.

/usr/share/doc/ansible-freeipa/playbooks/automember/ 디렉터리에 있는 automember-group-present.yml Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-group-present.yml automember-group-present-copy.yml
```

3.

편집할 automember-group-present-copy.yml 파일을 엽니다.

4.

ipaautomember 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin\_password 변수를 IdM 관리자 암호로 설정합니다.
- name 변수를 testing\_group 으로 설정합니다.
- automember\_type 변수를 그룹으로 설정합니다.
- state 변수가 present 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Automember group present example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure group automember rule admins is present
      ipaautomember:
        ipaadmin_password: Secret123
        name: testing_group
        automember_type: group
        state: present
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory automember-group-present-copy.yml
```

추가 리소스

- 자동 그룹 멤버십 및 **Automember** 규칙의 이점 을 참조하십시오.
- **Ansible**을 사용하여 **IdM** 사용자 그룹 **automember** 규칙에 조건이 있는지 확인합니다.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-automember.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/automember` 디렉토리를 참조하십시오.

### 16.3. **ANSIBLE**을 사용하여 **IDM** 사용자 그룹 **AUTOMEMBER** 규칙에 지정된 조건이 있는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM**(Identity Management) 그룹의 **automember** 규칙에 지정된 조건이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 **automember** 규칙에 **UID** 관련 조건이 있는 경우 **testing\_group** 그룹에 대해 확인됩니다. `.*` 조건을 지정하면 향후 모든 **IdM** 사용자가 자동으로 **testing\_group**의 멤버가 되도록 합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **IdM**에 **testing\_group** 사용자 그룹 및 **automember** 사용자 그룹 규칙이 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.

- `~/MyPlaybooks/` 디렉터리에서 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

## 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/automember/ directory`(예: `automember-usergroup-rule-present.yml`)에 있는 `automember-hostgroup-rule-present.yml` Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-present.yml automember-usergroup-rule-present.yml
```

3. 편집할 `automember-usergroup-rule-present.yml` 파일을 엽니다.

4. 다음 매개 변수를 수정하여 파일을 조정합니다.

- 플레이북의 이름을 사용 사례에 맞게 변경합니다(예: **Automember** 사용자 그룹 규칙 멤버).
- 사용 사례에 맞게 작업 이름을 변경합니다. 예를 들어 사용자 그룹의 **automember** 조건이 있는지 확인하십시오.
- **ipaautomember** 작업 섹션에서 다음 변수를 설정합니다.
  - **ipaadmin\_password** 변수를 **IdM** 관리자 암호로 설정합니다.
  - **name** 변수를 **testing\_group** 으로 설정합니다.
  - **automember\_type** 변수를 그룹으로 설정합니다.

- **state** 변수가 **present** 로 설정되어 있는지 확인합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
- **include** 키 변수 를 **UID** 로 설정합니다.
- 포함 식 변수를 **.\***로 설정합니다.\*

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Automember user group rule member present
  hosts: ipaserver
  become: true
  tasks:
  - name: Ensure an automember condition for a user group is present
    ipaautomember:
      ipaadmin_password: Secret123
      name: testing_group
      automember_type: group
      state: present
      action: member
      inclusive:
        - key: UID
          expression: .*
```

5. 파일을 저장합니다.

6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory automember-usergroup-rule-present.yml
```

## 검증 단계

1. **IdM** 관리자로 로그인합니다.

```
$ kinit admin
```

2.

다음과 같은 사용자를 추가합니다.

```
$ 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
...
```

추가 리소스

- [IdM CLI를 사용하여 기존 항목에 자동 멤버십 규칙 적용을 참조하십시오.](#)
- [자동 그룹 멤버십 및 Automember 규칙의 이점](#) 을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-automember.md 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/automember 디렉토리를 참조하십시오.](#)

#### 16.4. ANSIBLE을 사용하여 IDM 사용자 그룹 AUTOMEMBER 규칙에 조건이 없는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 그룹에 대한 자동 멤버 규칙에서 조건이 없는지 확인하는 방법을 설명합니다. 이 예제에서는 자동 멤버 규칙에 조건이 없으면 초기 사용자가 **dp** 여야 함을 지정하는지 확인합니다. **automember** 규칙이 **testing\_group** 그룹에 적용됩니다. 조건을 적용하면 초기 단계가 **dp** 인 향후 **IdM** 사용자가 **testing\_group** 의 멤버가 되지 않도록 합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **IdM**에 **testing\_group** 사용자 그룹 및 **automember** 사용자 그룹 규칙이 있습니다.

- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - `~/MyPlaybooks/` 디렉터리에서 이러한 옵션을 구성하는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

## 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/automember/` 디렉터리(예: `automember-usergroup-rule-absent.yml`)에 있는 `automember-hostgroup-rule-absent.yml` Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-absent.yml automember-usergroup-rule-absent.yml
```

3. 편집할 `automember-usergroup-rule-absent.yml` 파일을 엽니다.
4. 다음 매개 변수를 수정하여 파일을 조정합니다.

- 플레이북의 이름을 사용 사례에 맞게 변경합니다(예: **Automember** 사용자 그룹 규칙 멤버 **absent**).
- 사용 사례에 해당하는 작업 이름 변경(예: 사용자 그룹의 **automember** 조건 확인)이 없습니다.
- **ipaautomember** 작업 섹션에서 다음 변수를 설정합니다.

- **ipaadmin\_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 **testing\_group** 으로 설정합니다.
- **automember\_type** 변수를 그룹으로 설정합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
- **include** 키 변수 를 **initial** 로 설정합니다.
- 포함 식 변수를 **dp** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Automember user group rule member absent
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure an automember condition for a user group is absent
      ipaautomember:
        ipaadmin_password: Secret123
        name: testing_group
        automember_type: group
        state: absent
        action: member
        inclusive:
          - key: initials
            expression: dp
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory automember-usergroup-rule-absent.yml
```

#### 검증 단계

1. **IdM** 관리자로 로그인합니다.

```
$ kinit admin
```

2. **automember** 그룹을 확인합니다.

```
$ ipa automember-show --type=group testing_group
Automember Rule: testing_group
```

출력에 결정된 **Regex: initials=dp** 항목이 없으면 **testing\_group automember** 규칙에 지정된 조건이 포함되어 있지 않습니다.

#### 추가 리소스

- [IdM CLI를 사용하여 기존 항목에 자동 멤버십 규칙 적용을 참조하십시오.](#)
- [자동 그룹 멤버십 및 Automember 규칙의 이점](#) 을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-automember.md 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/automember 디렉토리를 참조하십시오.](#)

### 16.5. ANSIBLE을 사용하여 IDM 사용자 그룹의 AUTOMEMBER 규칙이 없는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 그룹에 대한 **automember** 규칙이 없는지 확인하는 방법을 설명합니다. 이 예제에서는 **testing\_group** 그룹에 대해 **automember** 규칙이 없는지 확인합니다.



## 참고

**automember** 규칙을 삭제하면 규칙과 관련된 모든 조건도 삭제됩니다. 규칙에서 특정 조건만 제거하려면 **Ansible**을 사용하여 **IdM** 사용자 그룹 **automember** 규칙에 조건이 없는지 확인합니다.

## 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - `~/MyPlaybooks/` 디렉터리에서 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

## 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/automember/` 디렉터리에 있는 **automember-group-absent.yml** Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-group-absent.yml automember-group-absent-copy.yml
```

3. 편집할 **automember-group-absent-copy.yml** 파일을 엽니다.

4. **ipaautomember** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 IdM 관리자 암호로 설정합니다.
- `name` 변수를 `testing_group` 으로 설정합니다.
- `automember_type` 변수를 그룹으로 설정합니다.
- `state` 변수가 `absent` 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Automember group absent example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure group automember rule admins is absent
      ipaautomember:
        ipaadmin_password: Secret123
        name: testing_group
        automember_type: group
        state: absent
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory automember-group-absent.yml
```

추가 리소스

- [자동 그룹 멤버십 및 Automember 규칙의 이점](#) 을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 `README-automember.md` 파일을 참조하십시오.

- `/usr/share/doc/ansible-freeipa/playbooks/automember` 디렉토리를 참조하십시오.

## 16.6. ANSIBLE을 사용하여 IDM 호스트 그룹 AUTOMEMBER 규칙에 조건이 있는지 확인합니다.

이 섹션에서는 Ansible을 사용하여 IdM 호스트 그룹 `automember` 규칙에 조건이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 FQDN 이 `*.idm.example.com` 인 호스트가 `primary_dns_domain_hosts` 호스트 그룹의 멤버이고 FQDN 이 `*.example.org` 가 `primary_dns_domain_hosts` 호스트 그룹의 멤버가 아닌지 확인하는 방법을 설명합니다.

### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- IdM에 `primary_dns_domain_hosts` 호스트 그룹과 `automember` 호스트 그룹 규칙이 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - `ansible-freeipa` 패키지를 설치했습니다.
  - `~/MyPlaybooks/` 디렉터리에서 이러한 옵션을 구성하는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

### 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/automember/` 디렉터리에 있는 `automember-hostgroup-rule-present.yml` Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-present.yml automember-hostgroup-rule-present-copy.yml
```

3.

편집할 **automember-hostgroup-rule-present-copy.yml** 파일을 엽니다.

4.

**ipaautomember** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 IdM 관리자 암호로 설정합니다.
- **name** 변수를 **primary\_dns\_domain\_hosts** 로 설정합니다.
- **automember\_type** 변수를 **hostgroup** 으로 설정합니다.
- **state** 변수가 **present** 로 설정되어 있는지 확인합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
- **inclusive** 키 변수가 **fqdn;**으로 설정되어 있는지 확인합니다.
- 해당하는 **include** 표현식 변수 를 **\*.idm.example.com** 으로 설정합니다.
- 전용 키 변수를 **fqdn;**로 설정합니다.
- 해당 전용 표현식 변수를 **\*.example.org** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Automember user group rule member present
  hosts: ipaserver
  become: true
  tasks:
```

```

- name: Ensure an automember condition for a user group is present
  ipaautomember:
    ipaadmin_password: Secret123
    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.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory automember-hostgroup-rule-present-copy.yml
```

#### 추가 리소스

- [IdM CLI를 사용하여 기존 항목에 자동 멤버십 규칙 적용을 참조하십시오.](#)
- [자동 그룹 멤버십 및 Automember 규칙의 이점](#) 을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-automember.md 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/automember 디렉토리를 참조하십시오.](#)

#### 16.7. 추가 리소스

- [Ansible 플레이북을 사용하여 사용자 계정 관리](#)
- [Ansible 플레이북을 사용하여 호스트 관리](#)

- **Ansible** 플레이북을 사용하여 사용자 그룹 관리
- **IdM CLI**를 사용하여 호스트 그룹 관리

## 17장. IDM CLI를 사용하여 사용자를 관리할 수 있도록 사용자 그룹에 권한 위임

위임은 셀프 서비스 규칙 및 **RBAC**(역할 기반 액세스 제어)와 함께 **IdM**의 액세스 제어 방법 중 하나입니다. 위임을 사용하여 한 사용자 그룹에 권한을 할당하여 다른 사용자 그룹의 항목을 관리할 수 있습니다.

이 섹션에서는 다음 주제를 다룹니다.

- [위임 규칙](#)
- [IdM CLI를 사용하여 위임 규칙 생성](#)
- [IdM CLI를 사용하여 기존 위임 규칙 보기](#)
- [IdM CLI를 사용하여 위임 규칙 수정](#)
- [IdM CLI를 사용하여 위임 규칙 삭제](#)

### 17.1. 위임 규칙

위임 규칙을 만들어 사용자를 관리하기 위해 사용자 그룹에 권한을 위임할 수 있습니다.

위임 규칙을 사용하면 특정 사용자 그룹에서 다른 사용자 그룹의 사용자에게 대한 특정 속성에 대한 쓰기 (편집) 작업을 수행할 수 있습니다. 이 형식의 액세스 제어 규칙은 위임 규칙에 지정한 속성의 하위 집합 값을 편집하도록 제한됩니다. 전체 항목을 추가하거나 제거하지 않은 속성을 제어하는 기능을 부여하지 않습니다.

위임 규칙은 **IdM**의 기존 사용자 그룹에 권한을 부여합니다. 예를 들어 **managers** 사용자 그룹이 **employees** 사용자 그룹에 있는 사용자의 선택된 특성을 관리할 수 있도록 위임을 사용할 수 있습니다.

### 17.2. IDM CLI를 사용하여 위임 규칙 생성

이 섹션에서는 **IdM CLI**를 사용하여 위임 규칙을 생성하는 방법을 설명합니다.

## 사전 요구 사항

- **admins** 그룹의 멤버로 로그인되어 있습니다.

## 절차

- **ipa delegation-add** 명령을 입력합니다. 다음 옵션을 지정합니다.
  - **--group:** 사용자 그룹의 사용자 항목에 대한 권한이 부여된 그룹입니다.
  - **--memberof:** 위임 그룹의 멤버가 항목을 편집할 수 있는 그룹입니다.
  - **--permissions:** 사용자가 지정된 특성을 볼 수 있는 권한이 있는지(읽기)와 지정된 특성을 추가하거나 변경할 수 있는지 여부(쓰기). 권한을 지정하지 않으면 쓰기 권한만 추가됩니다.
  - **--attrs:** 멤버 그룹의 사용자가 보거나 편집할 수 있는 속성입니다.

예를 들어 다음과 같습니다.

```
$ ipa delegation-add "basic manager attributes" --permissions=read --permissions=write --
attrs=businesscategory --attrs=departmentnumber --attrs=employeetype --
attrs=employeenumber --group=managers --memberof=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
```

## 17.3. IDM CLI를 사용하여 기존 위임 규칙 보기

이 섹션에서는 **IdM CLI**를 사용하여 기존 위임 규칙을 보는 방법을 설명합니다.

## 사전 요구 사항

- **admins** 그룹의 멤버로 로그인되어 있습니다.

#### 절차

- **ipa delegation-find** 명령을 입력합니다.

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

## 17.4. IDM CLI를 사용하여 위임 규칙 수정

이 섹션에서는 **IdM CLI**를 사용하여 기존 위임 규칙을 수정하는 방법을 설명합니다.

#### 중요

**--attrs** 옵션은 지원되는 속성의 이전 목록을 덮어쓰므로 항상 새 속성과 함께 전체 속성 목록을 포함합니다. 이는 **--permissions** 옵션에도 적용됩니다.

#### 사전 요구 사항

- **admins** 그룹의 멤버로 로그인되어 있습니다.

#### 절차

- 원하는 변경 사항을 사용하여 **ipa delegation-mod** 명령을 입력합니다. 예를 들어 기본 관리자 특성 예제 규칙에 **displayname** 특성을 추가하려면 다음을 수행합니다.

```
$ 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

```

## 17.5. IDM CLI를 사용하여 위임 규칙 삭제

이 섹션에서는 **IdM CLI**를 사용하여 기존 위임 규칙을 삭제하는 방법을 설명합니다.

### 사전 요구 사항

- **admins** 그룹의 멤버로 로그인되어 있습니다.

### 절차

- **ipa delegation-del** 명령을 입력합니다.
- 메시지가 표시되면 삭제할 위임 규칙의 이름을 입력합니다.

```

$ ipa delegation-del
Delegation name: basic manager attributes
-----
Deleted delegation "basic manager attributes"
-----

```

## 18장. IDM WEBUI를 사용하여 사용자를 관리할 수 있도록 사용자 그룹에 권한 위임

위임은 셀프 서비스 규칙 및 **RBAC**(역할 기반 액세스 제어)와 함께 **IdM**의 액세스 제어 방법 중 하나입니다. 위임을 사용하여 한 사용자 그룹에 권한을 할당하여 다른 사용자 그룹의 항목을 관리할 수 있습니다.

이 섹션에서는 다음 주제를 다룹니다.

- [위임 규칙](#)
- [IdM WebUI를 사용하여 위임 규칙 생성](#)
- [IdM WebUI를 사용하여 기존 위임 규칙 보기](#)
- [IdM WebUI를 사용하여 위임 규칙 수정](#)
- [IdM WebUI를 사용하여 위임 규칙 삭제](#)

### 18.1. 위임 규칙

위임 규칙을 만들어 사용자를 관리하기 위해 사용자 그룹에 권한을 위임할 수 있습니다.

위임 규칙을 사용하면 특정 사용자 그룹에서 다른 사용자 그룹의 사용자에게 대한 특정 속성에 대한 쓰기 (편집) 작업을 수행할 수 있습니다. 이 형식의 액세스 제어 규칙은 위임 규칙에 지정한 속성의 하위 집합 값을 편집하도록 제한됩니다. 전체 항목을 추가하거나 제거하지 않은 속성을 제어하는 기능을 부여하지 않습니다.

위임 규칙은 **IdM**의 기존 사용자 그룹에 권한을 부여합니다. 예를 들어 **managers** 사용자 그룹이 **employees** 사용자 그룹에 있는 사용자의 선택된 특성을 관리할 수 있도록 위임을 사용할 수 있습니다.

### 18.2. IDM WEBUI를 사용하여 위임 규칙 생성

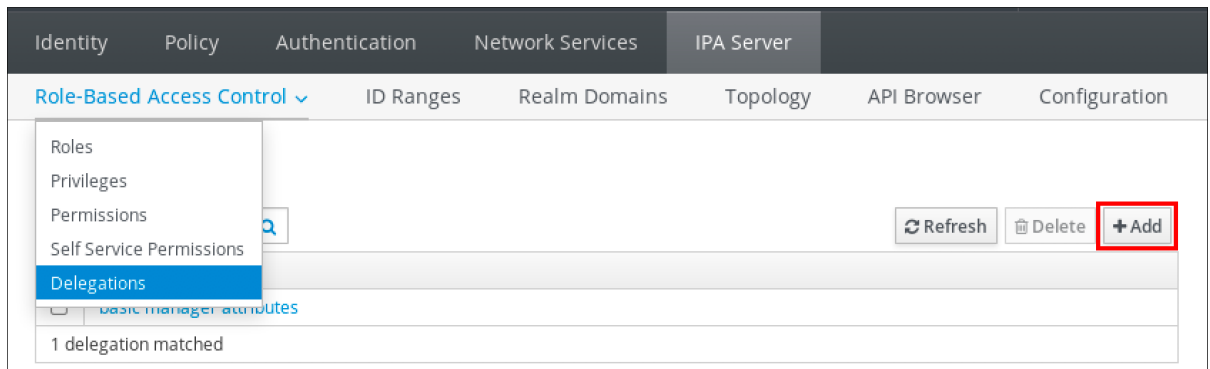
이 섹션에서는 **IdM WebUI**를 사용하여 위임 규칙을 생성하는 방법을 설명합니다.

## 사전 요구 사항

- IdM 웹 UI에 **admins** 그룹의 멤버로 로그인되어 있습니다.

## 절차

1. IPA 서버 메뉴에서 역할 기반 액세스 제어 → **Delegations** 를 클릭합니다.
2. 추가를 클릭합니다.



3. 위임 추가 창에서 다음을 수행합니다.
  - a. 새 위임 규칙의 이름을 지정합니다.
  - b. 사용자가 지정된 특성을 볼 수 있는지(읽기)하고 지정된 특성을 추가하거나 변경할 수 있는 권한을 나타내는 확인란을 선택하여 권한을 설정합니다(쓰기).
  - c. 사용자 그룹 드롭다운 메뉴에서 **권한이 부여되는 그룹**을 선택하여 멤버 그룹의 사용자 항목을 보거나 편집할 수 있습니다.
  - d. 멤버 사용자 그룹 드롭다운 메뉴에서 위임 그룹 멤버가 편집할 수 있는 항목을 선택합니다. **In the Member user group drop-down menu, select the group *whose entries can be edited* by members of the delegation group.**
  - e. 특성 상자에서 권한을 부여하려는 속성의 확인란을 선택합니다.

Add delegation

Delegation name \*
basic manager attributes

Permissions
☒ read
☒ write

User group \*
managers

Member user group \*
employees

Attributes \*

Filter
Add

☐ audio
☒ businesscategory

☐ carlicense
☐ cn

☒ departmentnumber
☐ description

☐ destinationindicator
☐ displayname

☒ employeeenumber
☒ employeetype

☐ facsimiletelephonenumber
☐ gecos

☐ gidnumber
☐ givenname

☐ homedirectory
☐ homephone

☐ homepostaladdress
☐ inetuserhttpurl

☐ inetuserstatus
☐ initials

☐ internationalisdnnumber
☐ ipacertmapdata

☐ ipakrbauthzdata
☐ ipanthash

☐ ipanthomedirectory
☐ ipanthomedirectorydrive

☐ ipantlogonscript
☐ ipantprofilepath

☐ ipantsecurityidentifier
☐ ipasshpubkey

☐ ipatokenradiusconfiglink
☐ ipatokenradiususername

☐ ipauniqueid
☐ ipauserauthtype

☐ jpegphoto
☐ krballowedtodelegateto

☐ krbcanonicalname
☐ krbextradata

\* Required field

Add

Add and Add Another

Add and Edit

Cancel

f.

**Add** 버튼을 클릭하여 새 위임 규칙을 저장합니다.

### 18.3. IDM WEBUI를 사용하여 기존 위임 규칙 보기

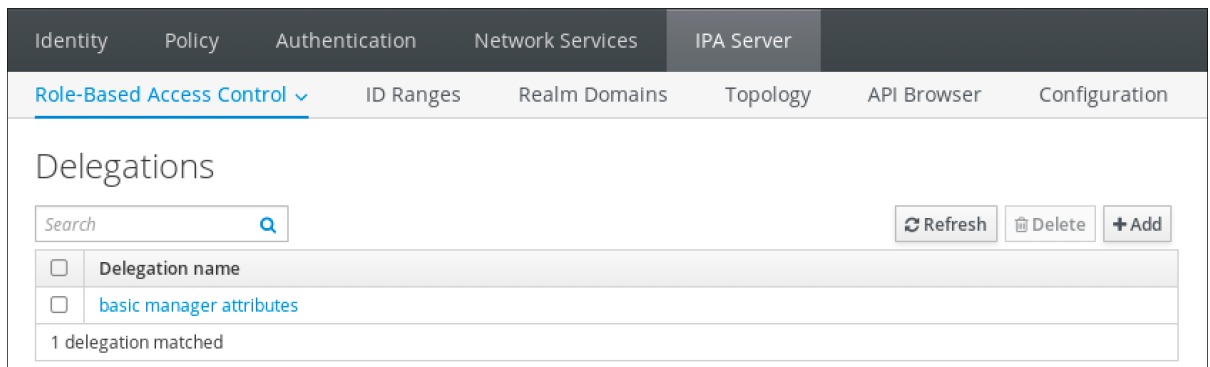
이 섹션에서는 **IdM WebUI**를 사용하여 기존 위임 규칙을 보는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM 웹 UI에 admins 그룹의 멤버로 로그인되어 있습니다.**

#### 절차

- **IPA 서버 메뉴에서 역할 기반 액세스 제어 → Delegations** 를 클릭합니다.



### 18.4. IDM WEBUI를 사용하여 위임 규칙 수정

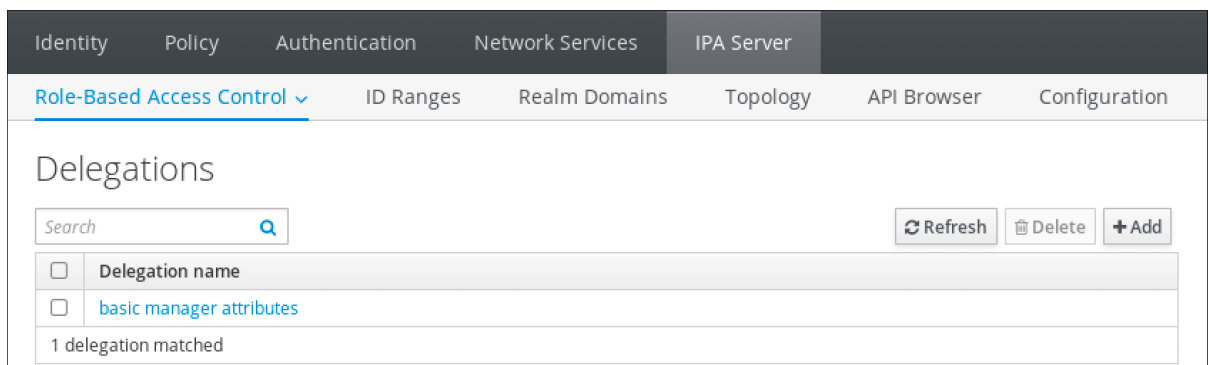
이 섹션에서는 **IdM WebUI**를 사용하여 기존 위임 규칙을 수정하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM 웹 UI에 admins 그룹의 멤버로 로그인되어 있습니다.**

#### 절차

1. **IPA 서버 메뉴에서 역할 기반 액세스 제어 → Delegations** 를 클릭합니다.



2.

수정할 규칙을 클릭합니다.

3.

원하는 대로 변경합니다.

•

규칙 이름을 변경합니다.

•

사용자가 지정된 특성을 볼 수 있는지 여부를 나타내는 확인란을 선택하고(읽기) 지정된 특성을 추가하거나 변경할 수 있는 확인란을 선택하여 부여된 권한을 변경합니다(쓰기).

•

사용자 그룹 드롭다운 메뉴에서 권한이 부여되는 그룹을 선택하여 멤버 그룹의 사용자 항목을 보거나 편집할 수 있습니다.

•

멤버 사용자 그룹 드롭다운 메뉴에서 위임 그룹 멤버가 편집할 수 있는 항목을 선택합니다. **In the Member user group drop-down menu, select the group *whose entries can be edited* by members of the delegation group.**

•

특성 상자에서 권한을 부여하려는 속성의 확인란을 선택합니다. 특성에 대한 권한을 제거하려면 관련 확인란을 선택 취소합니다.

Role-Based Access Control ▾ ID Ranges Realm Domains Topology API Browser Configuration

Delegations > basic manager attributes

Delegation: basic manager attributes

Settings

Refresh Revert **Save**

General

Delegation name basic manager attributes

Permissions \* ☒ read ☒ write

User group \* managers ▾

Member user group \* employees ▾

Attributes \*

|                                                  |                                                      |                                                     |
|--------------------------------------------------|------------------------------------------------------|-----------------------------------------------------|
| <input type="checkbox"/> audio                   | <input checked="" type="checkbox"/> businesscategory | <input type="checkbox"/> carlicense                 |
| <input type="checkbox"/> cn                      | <input checked="" type="checkbox"/> departmentnumber | <input type="checkbox"/> description                |
| <input type="checkbox"/> destinationindicator    | <input checked="" type="checkbox"/> displayname      | <input checked="" type="checkbox"/> employeeenumber |
| <input checked="" type="checkbox"/> employeetype | <input type="checkbox"/> facsimiletelephonenumber    | <input type="checkbox"/> gecoss                     |
| <input type="checkbox"/> gidnumber               | <input type="checkbox"/> givenname                   | <input checked="" type="checkbox"/> homedirectory   |
| <input type="checkbox"/> homephone               | <input type="checkbox"/> homepostaladdress           | <input type="checkbox"/> inetuserhttpurl            |
| <input type="checkbox"/> inetuserstatus          | <input type="checkbox"/> initials                    | <input type="checkbox"/> internationalisdnumber     |
| <input type="checkbox"/> ipacertmapdata          | <input type="checkbox"/> ipakrbauthzdata             | <input type="checkbox"/> ipanhash                   |
| <input type="checkbox"/> ipanthomedirectory      | <input type="checkbox"/> ipanthomedirectorydrive     | <input type="checkbox"/> ipantlogonscript           |
| <input type="checkbox"/> ipantprofilepath        | <input type="checkbox"/> ipantsecurityidentifier     | <input type="checkbox"/> ipasshpubkey               |

- 저장 버튼을 클릭하여 변경 사항을 저장합니다.

## 18.5. IDM WEBUI를 사용하여 위임 규칙 삭제

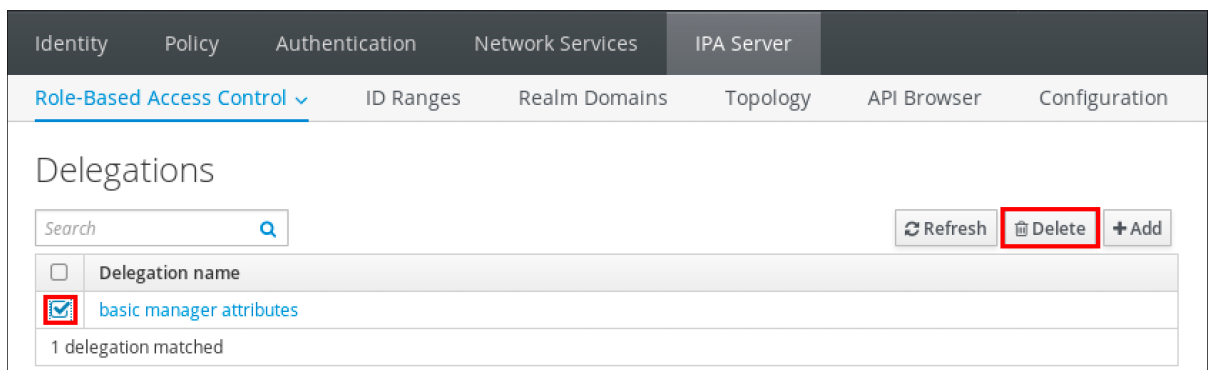
이 섹션에서는 **IdM WebUI**를 사용하여 기존 위임 규칙을 삭제하는 방법을 설명합니다.

### 사전 요구 사항

- **IdM 웹 UI에 admins 그룹의 멤버로 로그인되어 있습니다.**

### 절차

1. **IPA 서버 메뉴에서 역할 기반 액세스 제어 → Delegations**를 클릭합니다.
2. 제거할 규칙 옆에 있는 확인란을 선택합니다.
3. 삭제를 클릭합니다.



4. **Delete**를 클릭하여 확인합니다.

## 19장. ANSIBLE 플레이북을 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한 위임

위임은 셀프 서비스 규칙 및 **RBAC**(역할 기반 액세스 제어)와 함께 **IdM**의 액세스 제어 방법 중 하나입니다. 위임을 사용하여 한 사용자 그룹에 권한을 할당하여 다른 사용자 그룹의 항목을 관리할 수 있습니다.

이 섹션에서는 다음 주제를 다룹니다.

- [위임 규칙](#)
- [IdM용 Ansible 인벤토리 파일 생성](#)
- [Ansible을 사용하여 위임 규칙이 있는지 확인합니다.](#)
- [Ansible을 사용하여 위임 규칙이 없는지 확인합니다.](#)
- [Ansible을 사용하여 위임 규칙에 특정 속성이 있는지 확인](#)
- [Ansible을 사용하여 위임 규칙에 특정 속성이 없는지 확인](#)

### 19.1. 위임 규칙

위임 규칙을 만들어 사용자를 관리하기 위해 사용자 그룹에 권한을 위임할 수 있습니다.

위임 규칙을 사용하면 특정 사용자 그룹에서 다른 사용자 그룹의 사용자에게 대한 특정 속성에 대한 쓰기 (편집) 작업을 수행할 수 있습니다. 이 형식의 액세스 제어 규칙은 위임 규칙에 지정한 속성의 하위 집합 값을 편집하도록 제한됩니다. 전체 항목을 추가하거나 제거하지 않은 속성을 제어하는 기능을 부여하지 않습니다.

위임 규칙은 **IdM**의 기존 사용자 그룹에 권한을 부여합니다. 예를 들어 **managers** 사용자 그룹이 **employees** 사용자 그룹에 있는 사용자의 선택된 특성을 관리할 수 있도록 위임을 사용할 수 있습니다.

### 19.2. IDm용 ANSIBLE 인벤토리 파일 생성

**Ansible**을 사용하여 작업하는 경우 홈 디렉토리에서 `/usr/share/doc/ansible-freeipa/*` 및 `/usr/share/doc/rhel-system-roles/*` 하위 디렉토리에서 복사하여 조정하는 하위 디렉토리를 생성하는 것이 좋습니다. 이 방법은 다음과 같은 장점이 있습니다.

- 모든 플레이북을 한 곳에서 찾을 수 있습니다.
- **root** 권한을 호출하지 않고 플레이북을 실행할 수 있습니다.

## 절차

1. 홈 디렉토리에 **Ansible** 구성 및 플레이북의 디렉토리를 생성합니다.

```
$ mkdir ~/MyPlaybooks/
```

2. `~/MyPlaybooks/` 디렉토리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3. 다음 콘텐츠를 사용하여 `~/MyPlaybooks/ansible.cfg` 파일을 생성합니다.

```
[defaults]
inventory = /home/<username>/MyPlaybooks/inventory

[privilege_escalation]
become=True
```

4. 다음 콘텐츠를 사용하여 `~/MyPlaybooks/inventory` 파일을 만듭니다.

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

이 구성은 해당 위치에 있는 호스트에 대해 **eu** 및 **us**의 두 개의 호스트 그룹을 정의합니다. 또

한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트가 포함된 **ipaserver** 호스트 그룹을 정의합니다.

### 19.3. ANSIBLE을 사용하여 위임 규칙이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 새 **IdM** 위임 규칙에 대한 권한을 정의하고 있는지 확인하는 방법을 설명합니다. 이 예제에서 새 기본 관리자 속성 위임 규칙은 **managers** 그룹에 **staff** 그룹의 멤버에 대해 다음 속성을 읽고 쓸 수 있는 기능을 부여합니다.

- **businesscategory**
- **departmentnumber**
- **employeenumber**
- **employeetype**

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
  - **Ansible** 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.

#### 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/delegation/` 디렉터리에 있는 `delegation-present.yml` 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-present.yml
delegation-present-copy.yml
```

3. 편집할 `delegation-present-copy.yml` Ansible 플레이북 파일을 엽니다.

4. `ipadelegation` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- `name` 변수를 새 위임 규칙의 이름으로 설정합니다.
- 권한 변수를 쉼표로 구분된 권한 목록으로 설정하여 부여할 권한(읽기 및 쓰기)을 부여합니다.
- 특성 변수를 위임된 사용자 그룹이 관리할 수 있는 속성 목록으로 설정합니다. `businesscategory, departmentnumber, staffnumber, and employeetype`.
- 특성을 보거나 수정할 수 있도록 액세스 권한이 부여되는 그룹의 이름으로 그룹 변수를 설정합니다.
- `membergroup` 변수를 보거나 수정할 수 있는 그룹의 이름으로 설정합니다.

현재 예제에 대해 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to manage a delegation rule
  hosts: ipaserver
```

```
become: true
```

```
tasks:
```

```
- name: Ensure delegation "basic manager attributes" is present
```

```
  ipadelegation:
```

```
    ipaadmin_password: Secret123
```

```
    name: "basic manager attributes"
```

```
    permission: read, write
```

```
    attribute:
```

```
      - businesscategory
```

```
      - departmentnumber
```

```
      - employeenumber
```

```
      - employeetype
```

```
    group: managers
```

```
    membergroup: employees
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-present-copy.yml
```

추가 리소스

- 

분류 [규칙](#)을 참조하십시오.

- 

`/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.

- 

`/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉터리에서 샘플 플레이북을 참조하십시오.

#### 19.4. ANSIBLE을 사용하여 위임 규칙이 없는지 확인합니다.

다음 절차에서는 **IdM** 구성에 지정된 위임 규칙이 없는지 확인하는 데 **Ansible** 플레이북을 사용하는 방법을 설명합니다. 아래 예제에서는 **IdM**에 사용자 정의 기본 관리자 속성 위임 규칙이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - 이러한 옵션을 구성하는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
  - **Ansible** 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.

#### 절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/delegation/** 디렉터리에 있는 **delegation-absent.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-present.yml  
delegation-absent-copy.yml
```

3. 편집할 **delegation-absent-copy.yml** **Ansible** 플레이북 파일을 엽니다.
4. **ipadelegation** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 IdM 관리자의 암호로 설정합니다.
- **name** 변수를 위임 규칙의 이름으로 설정합니다.

- **state** 변수를 **absent** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Delegation absent
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure delegation "basic manager attributes" is absent
      ipadelegation:
        ipaadmin_password: Secret123
        name: "basic manager attributes"
        state: absent
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-absent-copy.yml
```

추가 리소스

- 분류 [규칙](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉터리에서 샘플 플레이북을 참조하십시오.

## 19.5. ANSIBLE을 사용하여 위임 규칙에 특정 속성이 있는지 확인

다음 절차에서는 위임 규칙에 특정 설정이 있도록 **Ansible** 플레이북을 사용하는 방법을 설명합니다. 이 플레이북을 사용하여 이전에 만든 위임 역할을 수정할 수 있습니다. 이 예제에서는 기본 관리자 속성 위임 규칙만 **departmentnumber member** 특성을 갖도록 합니다. In the example, you ensure the basic manager attributes delegation rule only has the departmentnumber member attribute.

## 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - 이러한 옵션을 구성하는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
  - **Ansible** 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.
- IdM에 기본 관리자 속성 의 위임 규칙이 있습니다.

## 절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.
 

```
$ cd ~/MyPlaybooks/
```
2. **/usr/share/doc/ansible-freeipa/playbooks/delegation/** 디렉터리에 있는 **delegation-member-present.yml** 파일의 사본을 만듭니다.
 

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-member-present.yml delegation-member-present-copy.yml
```
3. 편집할 **delegation-member-present-copy.yml** **Ansible** 플레이북 파일을 엽니다.
4. **ipadelegation** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 수정할 위임 규칙의 이름으로 설정합니다.
- 특정 변수를 **departmentnumber** 로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Delegation member present
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure delegation "basic manager attributes" member attribute
      departmentnumber is present
      ipadelegation:
        ipaadmin_password: Secret123
        name: "basic manager attributes"
        attribute:
          - departmentnumber
        action: member
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-member-present-copy.yml
```

#### 추가 리소스

- 분류 [규칙](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.

- `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉토리에서 샘플 플레이북을 참조하십시오.

## 19.6. ANSIBLE을 사용하여 위임 규칙에 특정 속성이 없는지 확인

다음 절차에서는 위임 규칙에 특정 설정이 없도록 **Ansible** 플레이북을 사용하는 방법을 설명합니다. 이 플레이북을 사용하여 위임 역할에 바람직하지 않은 액세스 권한이 부여되지 않도록 할 수 있습니다. 이 예제에서는 기본 관리자 속성 의 위임 규칙에 **staff number** 및 **employee type** 멤버 속성이 없는지 확인합니다.

### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
  - **ansible-freeipa** 패키지를 설치했습니다.
  - 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
  - **Ansible** 인벤토리 파일은 `~/MyPlaybooks/` 디렉터리에 있습니다.
- **IdM**에 기본 관리자 속성 의 위임 규칙이 있습니다.

### 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2.

`/usr/share/doc/ansible-freeipa/playbooks/delegation/` 디렉터리에 있는 `delegation-member-absent.yml` 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-member-absent.yml delegation-member-absent-copy.yml
```

3.

편집할 `delegation-member-absent-copy.yml` Ansible 플레이북 파일을 엽니다.

4.

`ipadelegation` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 `IdM` 관리자의 암호로 설정합니다.
- `name` 변수를 수정할 위임 규칙의 이름으로 설정합니다.
- 특정 변수를 `worker number` 및 `employee type` 로 설정합니다.
- `action` 변수를 `member` 로 설정합니다.
- `state` 변수를 `absent` 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Delegation member absent
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure delegation "basic manager attributes" member attributes
      employee number and employee type are absent
      ipadelegation:
        ipaadmin_password: Secret123
        name: "basic manager attributes"
        attribute:
          - employee number
          - employee type
        action: member
        state: absent
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-member-absent-copy.yml
```

#### 추가 리소스

- 분류 [규칙](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉터리에서 샘플 플레이북을 참조하십시오.

## 20장. CLI를 사용하여 IDM에서 역할 기반 액세스 제어 관리

이 장에서는 **IdM(Identity Management)**에서 역할 기반 액세스 제어를 소개하고 **CLI(명령줄 인터페이스)**에서 다음 작업을 설명합니다.

- 권한 관리
- 권한 관리
- 역할 관리

### 20.1. IDM의 역할 기반 액세스 제어

**IdM의 RBAC(역할 기반 액세스 제어)**는 셀프 서비스 및 위임 액세스 제어에 비해 사용자에게 매우 다른 종류의 권한을 부여합니다.

역할 기반 액세스 제어는 다음 세 부분으로 구성됩니다.

- 권한은 사용자 추가 또는 삭제, 그룹 수정, 읽기 액세스 등의 특정 작업을 수행할 수 있는 권한을 부여합니다.
- 권한은 예를 들어 새 사용자를 추가하는 데 필요한 모든 권한을 결합합니다.
- 역할에 따라 사용자, 사용자 그룹, 호스트 또는 호스트 그룹에 일련의 권한이 부여됩니다.

#### 20.1.1. IdM의 권한

권한은 역할 기반 액세스 제어의 가장 낮은 수준 단위이며 해당 작업이 적용되는 **LDAP** 항목과 함께 작업을 정의합니다. 빌딩 블록에 비해 필요한 만큼의 권한에 권한을 할당할 수 있습니다. 하나 이상의 권한은 허용되는 작업을 정의합니다.

- write

- **read**
- **search**
- **비교**
- **add**
- **delete**
- **all**

이러한 작업은 다음 세 가지 기본 대상에 적용됩니다.

- **subtree:** 도메인 이름(DN); 이 DN 아래의 하위 트리
- **대상 필터:** LDAP 필터
- **target:** 가능한 와일드카드가 있는 DN을 사용하여 항목을 지정

또한 다음 편의 옵션에서 해당 속성을 설정합니다.

- **type:** 개체 유형 (사용자, 그룹 등)은 하위 트리 와 대상 필터를 설정합니다.
- **memberOf:** 그룹의 멤버이며, 대상 필터를 설정합니다.
- **TargetGroup :** 특정 그룹을 수정할 수 있는 액세스 권한을 부여합니다(예: 그룹 멤버십을 관리할 수 있는 권한 부여)

**IdM** 권한을 사용하면 어떤 오브젝트에 액세스할 수 있는 사용자와 이러한 오브젝트의 속성을 제어할 수 있습니다. **IdM**을 사용하면 개별 속성을 허용 또는 차단하거나 사용자, 그룹 또는 **sudo**와 같은 특정 **IdM** 기능의 전체 가시성을 모든 익명 사용자, 인증된 모든 사용자 또는 특정 권한 있는 사용자 그룹에 변경할 수 있습니다.

예를 들어, 권한에 대한 이 접근 방식의 유연성은 사용자 또는 그룹의 액세스 권한을 이러한 사용자 또는 그룹에 대한 액세스만 제한하려는 관리자에게 유용합니다. 이러한 사용자 또는 그룹에 액세스해야 하며 다른 섹션을 완전히 숨겨야 합니다.



참고

권한은 다른 권한을 포함할 수 없습니다.

### 20.1.2. 기본 관리 권한

관리 권한은 기본적으로 **IdM**과 함께 제공되는 권한입니다. 이러한 권한은 사용자가 생성한 다른 권한처럼 작동하며 다음과 같은 차이점이 있습니다.

- 해당 항목을 삭제하거나 이름, 위치 및 대상 속성을 수정할 수 없습니다.
- 세 가지 속성 세트가 있습니다.
  - 기본 속성, 사용자가 **IdM**에서 관리하므로 수정할 수 없습니다.
  - 사용자가 추가한 추가 속성인 포함 특성
  - 사용자가 제거된 속성인 제외된 특성

관리 권한은 **default** 및 **included** 특성 세트에 표시되지만 제외 세트에는 표시되지 않는 모든 특성에 적용됩니다.



참고

관리 권한을 삭제할 수는 없지만 해당 **bind** 유형을 권한으로 설정하고 모든 권한에서 관리 권한을 제거하면 이를 효과적으로 비활성화합니다.

모든 관리 권한의 이름은 **System:** (예: **System: Add Sudo rule** or **System: Modify Services**)로 시작합니다. 이전 버전의 IdM에서는 기본 권한에 다른 스키마를 사용했습니다. 예를 들어 사용자는 해당 항목을 삭제할 수 없으며 해당 사용자를 권한에만 할당할 수 있었습니다. 이러한 기본 권한의 대부분은 관리 되는 권한으로 전환되었지만 다음 권한에서는 여전히 이전 스키마를 사용합니다.

- 멤버십 자동 작성 작업 추가
- 설정 하위 항목 추가
- 복제 계약 추가
- 인증서 제거 **Hold**
- CA에서 인증서 상태 가져오기
- DNA 범위 읽기
- DNA 범위 수정
- PassSync Managers 설정 읽기
- PassSync Managers 설정 수정
- 복제 계약 읽기
- 복제 계약 수정
- 복제 계약 제거

- **LDBM 데이터베이스 구성 읽기**
- 요청 인증서
- **CA ACL을 무시하는 인증서 요청**
- 다른 호스트의 인증서 요청
- **CA에서 인증서 검색**
- 인증서 해지
- **IPA 설정 쓰기**



#### 참고

명령줄에서 관리 권한을 수정하려고 하면 시스템에서 수정할 수 없는 속성을 변경할 수 없으므로 명령이 실패합니다. 웹 UI에서 관리되는 권한을 수정하려고 하면 수정할 수 없는 특성이 비활성화됩니다.

### 20.1.3. IdM의 권한

권한은 역할에 적용 가능한 권한 그룹입니다.

권한은 단일 작업을 수행할 수 있는 권한을 제공하지만 성공하려면 여러 권한이 필요한 특정 IdM 작업이 있습니다. 따라서 권한은 특정 작업을 수행하는 데 필요한 다양한 권한을 결합합니다.

예를 들어 새 IdM 사용자에게 대한 계정을 설정하려면 다음 권한이 필요합니다.

- 새 사용자 항목 만들기
- 사용자 암호 재설정
- 기본 IPA 사용자 그룹에 새 사용자 추가

이러한 세 가지 하위 수준 작업을 라는 사용자 지정 권한 형식의 상위 수준 작업으로 결합하면 시스템 관리자가 역할을 보다 쉽게 관리할 수 있습니다. **IdM**에는 이미 여러 기본 권한이 포함되어 있습니다. 사용자 및 사용자 그룹 외에도 호스트 및 호스트 그룹과 네트워크 서비스에 권한이 할당됩니다. 이 방법을 사용하면 특정 네트워크 서비스를 사용하는 호스트 집합에서 일련의 사용자별로 작업을 세부적으로 제어할 수 있습니다.



#### 참고

권한에는 다른 권한이 포함되지 않을 수 있습니다.

### 20.1.4. IdM의 역할

역할은 역할에 지정된 사용자가 보유한 권한 목록입니다. 실제로 권한은 낮은 수준의 작업(사용자 항목 생성, 그룹에 항목을 추가 등)을 수행할 수 있는 기능을 부여합니다. 권한은 상위 수준의 작업(예: 지정된 그룹에서 새 사용자 생성)에 필요한 이러한 권한 중 하나 이상을 결합합니다. 역할은 필요에 따라 권한을 함께 수집합니다. 예를 들어 사용자 관리자 역할은 사용자를 추가, 수정, 삭제할 수 있습니다.



#### 중요

역할은 허용된 작업을 분류하는 데 사용됩니다. 권한 분리를 구현하거나 권한 에스컬레이션으로부터 보호하는 도구로는 사용되지 않습니다.



#### 참고

역할에 다른 역할을 포함할 수 없습니다.

### 20.1.5. ID 관리에서 사전 정의된 역할

**Red Hat Identity Management**는 다음과 같은 다양한 사전 정의 역할을 제공합니다.

표 20.1. ID 관리에서 사전 정의된 역할

| Role     | 권한                         | 설명                  |
|----------|----------------------------|---------------------|
| 등록 관리자   | 호스트 등록                     | 클라이언트 또는 호스트 등록 담당자 |
| helpdesk | 사용자 수정 및 암호 재설정, 그룹 멤버십 수정 | 간단한 사용자 관리 작업 수행    |

| Role      | 권한                                       | 설명                                 |
|-----------|------------------------------------------|------------------------------------|
| IT 보안 전문가 | Netgroups 관리자, HBAC 관리자, Sudo 관리자        | 호스트 기반 액세스 제어와 같은 보안 정책 관리 sudo 규칙 |
| IT 전문가    | 호스트 관리자, 호스트 그룹 관리자, 서비스 관리자, 자동 마운트 관리자 | 호스트 관리                             |
| 보안 아키텍트   | 위임 관리자, 복제 관리자, 쓰기 IPA 구성, 암호 정책 관리자     | ID 관리 환경 관리, 신뢰 생성, 복제 계약 생성       |
| 사용자 관리자   | 사용자 관리자, 그룹 관리자, 사용자 관리자 단계              | 사용자 및 그룹 생성                        |

## 20.2. CLI에서 IDM 권한 관리

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 IdM(Identity Management) 권한을 관리하는 방법을 설명합니다.

### 사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조](#) 하십시오.

### 절차

1. **ipa permission-add** 명령을 사용하여 새 권한 항목을 생성합니다.  
예를 들어 **dns admin** 이라는 권한을 추가하려면 다음을 수행합니다.

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

2. 다음 옵션을 사용하여 권한의 속성을 지정합니다.

- **--bindtype** 은 바인딩 규칙 유형을 지정합니다. 이 옵션은 모든, **anonymous** 및 권한 인수를 허용합니다. 권한 **bindtype**은 역할을 통해 이 권한을 부여한 사용자만 이 권한을 행사할 수 있음을 의미합니다.  
예를 들어 다음과 같습니다.

```
$ ipa permission-add "dns admin" --bindtype=all
```

**--bindtype** 을 지정하지 않으면 **permission** 이 기본값입니다.



참고

기본이 아닌 바인딩 규칙 유형의 권한을 권한에 추가할 수 없습니다. 또한 기본이 아닌 바인딩 규칙 유형에 이미 있는 권한도 설정할 수 없습니다.

•

**--right** 는 권한에서 부여된 권한을 나열하며, 더 이상 사용되지 않는 **--permissions** 옵션을 대체합니다. 사용 가능한 값은 **add,delete,read,search,compare,write,all** 입니다.

여러 **--right** 옵션을 사용하거나 중괄호 안에 쉼표로 구분된 목록을 사용하여 여러 속성을 설정할 수 있습니다. 예를 들어 다음과 같습니다.

```
$ ipa permission-add "dns admin" --right=read --right=write
```

```
$ ipa permission-add "dns admin" --right={read,write}
```



참고

**add** 및 **delete** 는 사용자 삭제(예: 사용자 삭제, 그룹 추가 등)입니다. ,검색,비교 및 쓰기 는 더 많은 특성 수준: **userCertificate** 에 쓸 수 있지만 **userPassword** 를 읽을 수는 없습니다.

•

**--attrs** 는 권한이 부여된 속성 목록을 제공합니다.

여러 **--attrs** 옵션을 사용하거나 중괄호 안에 쉼표로 구분된 목록에 옵션을 나열하여 여러 속성을 설정할 수 있습니다. 예를 들어 다음과 같습니다.

```
$ ipa permission-add "dns admin" --attrs=description --attrs=automountKey
```

```
$ ipa permission-add "dns admin" --attrs={description,automountKey}
```

**--attrs** 와 함께 제공되는 속성이 있어야 하며 지정된 오브젝트 유형에 대해 허용되는 속성이 있어야 합니다. 그렇지 않으면 명령이 스키마 구문 오류로 인해 실패합니다.

•

**--type** 은 권한이 적용되는 항목 오브젝트 유형(예: **user**, **host** 또는 **service**)을 정의합니다. 각 유형에는 고유한 허용 속성 세트가 있습니다.

예를 들어 다음과 같습니다.

```
$ ipa permission-add "manage service" --right=all --type=service --
attrs=krbprincipalkey --attrs=krbprincipalname --attrs=managedby
```

•

**--subtree** 는 하위 트리 항목을 제공합니다. 필터는 이 하위 트리 항목 아래의 모든 항목을 대상으로 합니다. 기존 하위 트리 항목을 제공합니다. **--subtree** 는 와일드카드 또는 존재하지 않는 도메인 이름(DN)을 허용하지 않습니다. 디렉터리에 DN을 포함합니다.

IdM은 단순화된 플랫폼 디렉터리 트리 구조를 사용하므로 **--subtree** 를 사용하여 다른 구성의 컨테이너 또는 상위 항목인 자동 마운트 위치와 같은 일부 유형의 항목을 대상으로 할 수 있습니다. 예를 들어 다음과 같습니다.

```
$ 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
```



참고

**--type** 및 **--subtree** 옵션은 상호 배타적입니다. **--type** 에 대한 필터가 **--subtree** 의 간소화로 표시될 수 있으므로 관리자가 수월하기 쉽도록 합니다.

•

**--filter** 는 LDAP 필터를 사용하여 권한이 적용되는 항목을 식별합니다.

IdM은 지정된 필터의 유효성을 자동으로 확인합니다. 필터는 유효한 LDAP 필터일 수 있습니다. 예를 들면 다음과 같습니다.

```
$ ipa permission-add "manage Windows groups" --filter="(!
(objectclass=posixgroup))" --right=write --attrs=description
```

•

**--memberOf** 는 그룹이 존재하는지 확인한 후 지정된 그룹의 멤버로 **target** 필터를 설정합니다. 예를 들어, 이 권한이 있는 사용자가 **engineers** 그룹 멤버의 로그인 셸을 수정하도록 하려면 다음을 수행합니다.

```
$ ipa permission-add ManageShell --right="write" --type=user --attr=loginshell --
memberof=engineers
```

•

**--TargetGroup**은 그룹이 존재하는지 확인한 후 지정된 사용자 그룹에 **target**을 설정합니다. 예를 들어, 권한을 가진 사용자가 **engineers** 그룹에 **member** 특성을 쓰도록 하려면 (관리자는 멤버를 추가 또는 제거할 수 있음)

```
$ ipa permission-add ManageMembers --right="write" --
subtree=cn=groups,cn=accounts,dc=example,dc=test --attr=member --
targetgroup=engineers
```

- 선택적으로 대상 도메인 이름(DN)을 지정할 수 있습니다.
  - **--target** 은 권한을 적용할 DN을 지정합니다. 와일드카드가 허용됩니다.
  - **--targetto** 는 항목을 이동할 수 있는 DN 하위 트리를 지정합니다.
  - **--targetfrom** 은 항목을 이동할 수 있는 DN 하위 트리를 지정합니다.

### 20.3. 기존 권한에 대한 명령 옵션

다음 변형을 사용하여 필요에 따라 기존 권한을 수정합니다.

- 기존 권한을 편집하려면 **ipa permission-mod** 명령을 사용합니다. 와 동일한 명령 옵션을 사용하여 권한을 추가할 수 있습니다.
- 기존 권한을 찾으려면 **ipa permission-find** 명령을 사용합니다. 와 동일한 명령 옵션을 사용하여 권한을 추가할 수 있습니다.
- 특정 권한을 보려면 **ipa permission-show** 명령을 사용합니다.  
**--raw** 인수는 생성된 원시 **389-ds** ACI를 보여줍니다. 예를 들어 다음과 같습니다.

```
$ ipa permission-show <permission> --raw
```

- **ipa permission-del** 명령은 권한을 완전히 삭제합니다.

추가 리소스

- **ipa man** 페이지를 참조하십시오.

- **ipa help** 명령을 참조하십시오.

## 20.4. CLI에서 IDM 권한 관리

이 섹션에서는 **CLI**(명령줄 인터페이스)를 사용하여 **IdM**(Identity Management) 권한을 관리하는 방법을 설명합니다.

### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조](#) 하십시오.
- 기존 권한. 권한에 대한 자세한 내용은 [CLI의 IdM 권한](#) 관리를 참조하십시오.

### 절차

1. **ipa privilege-add** 명령을 사용하여 권한 항목을 추가하여 설명과 함께 *파일 시스템 관리* 라는 권한을 추가합니다.

```
$ ipa privilege-add "managing filesystems" --desc="for filesystems"
```

2. 예를 들어, **mount** 및 *managing ftp* 서비스를 관리하는 권한을 *파일 시스템* 권한에 추가하고, **privilege-add-permission** 명령을 사용하여 필요한 권한을 권한 그룹에 할당합니다.

```
$ ipa privilege-add-permission "managing filesystems" --permissions="managing automount" --permissions="managing ftp services"
```

## 20.5. 기존 권한에 대한 명령 옵션

다음 변형을 사용하여 필요에 따라 기존 권한을 수정합니다.

- 기존 권한을 수정하려면 **ipa privilege-mod** 명령을 사용합니다.

- 기존 권한을 찾으려면 **ipa privilege-find** 명령을 사용합니다.
- 특정 권한을 보려면 **ipa privilege-show** 명령을 사용합니다.
- **ipa privilege-remove-permission** 명령은 권한에서 하나 이상의 권한을 제거합니다.
- **ipa privilege-del** 명령은 권한을 완전히 삭제합니다.

#### 추가 리소스

- **ipa man** 페이지를 참조하십시오.
- **ipa help** 명령을 참조하십시오.

## 20.6. CLI에서 IDM 역할 관리

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 IdM(Identity Management) 역할을 관리하는 방법을 설명합니다.

#### 사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조하십시오](#).
- 기존 권한. 권한에 대한 자세한 내용은 [CLI의 IdM 권한 관리](#)를 참조하십시오.

#### 절차

1. **ipa role-add** 명령을 사용하여 새 역할 항목을 추가합니다.

```
$ ipa role-add --desc="User Administrator" useradmin
```

```
-----
Added role "useradmin"
-----
```

```
Role name: useradmin
```

```
Description: User Administrator
```

2.

`ipa role-add-privilege` 명령을 사용하여 역할에 필요한 권한을 추가합니다.

```
$ ipa role-add-privilege --privileges="user administrators" useradmin
```

```
Role name: useradmin
```

```
Description: User Administrator
```

```
Privileges: user administrators
```

```
-----
Number of privileges added 1
-----
```

3.

`ipa role-add-member` 명령을 사용하여 필요한 멤버를 역할에 추가합니다. 허용되는 멤버 유형은 사용자, 그룹, 호스트 및 호스트 그룹입니다.

예를 들어 **useradmins** 라는 그룹을 이전에 만든 **useradmin** 역할에 추가하려면 다음을 실행합니다.

```
$ ipa role-add-member --groups=useradmins useradmin
```

```
Role name: useradmin
```

```
Description: User Administrator
```

```
Member groups: useradmins
```

```
Privileges: user administrators
```

```
-----
Number of members added 1
-----
```

## 20.7. 기존 역할에 대한 명령 옵션

다음 변형을 사용하여 필요에 따라 기존 역할을 수정합니다.

•

기존 역할을 수정하려면 `ipa role-mod` 명령을 사용합니다.

•

기존 역할을 찾으려면 `ipa role-find` 명령을 사용합니다.

•

특정 역할을 보려면 `ipa role-show` 명령을 사용합니다.

•

역할에서 멤버를 제거하려면 `ipa role-remove-member` 명령을 사용합니다.

- **ipa role-remove-privilege** 명령은 역할에서 하나 이상의 권한을 제거합니다.
- **ipa role-del** 명령은 역할을 완전히 삭제합니다.

#### 추가 리소스

- **ipa man** 페이지 참조
- **ipa help** 명령을 참조하십시오.

## 21장. IDM 웹 UI를 사용하여 역할 기반 액세스 제어 관리

이 장에서는 **IdM(Identity Management)**에서 역할 기반 액세스 제어를 소개하고 웹 인터페이스(웹 UI)에서 다음 작업을 설명합니다.

- 권한 관리
- 권한 관리
- 역할 관리

### 21.1. IDM의 역할 기반 액세스 제어

**IdM의 RBAC(역할 기반 액세스 제어)**는 셀프 서비스 및 위임 액세스 제어에 비해 사용자에게 매우 다른 종류의 권한을 부여합니다.

역할 기반 액세스 제어는 다음 세 부분으로 구성됩니다.

- 권한은 사용자 추가 또는 삭제, 그룹 수정, 읽기 액세스 등의 특정 작업을 수행할 수 있는 권한을 부여합니다.
- 권한은 예를 들어 새 사용자를 추가하는 데 필요한 모든 권한을 결합합니다.
- 역할에 따라 사용자, 사용자 그룹, 호스트 또는 호스트 그룹에 일련의 권한이 부여됩니다.

#### 21.1.1. IdM의 권한

권한은 역할 기반 액세스 제어의 가장 낮은 수준 단위이며 해당 작업이 적용되는 **LDAP** 항목과 함께 작업을 정의합니다. 빌딩 블록에 비해 필요한 만큼의 권한에 권한을 할당할 수 있습니다. 하나 이상의 권한은 허용되는 작업을 정의합니다.

- write

- **read**
- **search**
- **비교**
- **add**
- **delete**
- **all**

이러한 작업은 다음 세 가지 기본 대상에 적용됩니다.

- **subtree:** 도메인 이름(DN); 이 DN 아래의 하위 트리
- **대상 필터:** LDAP 필터
- **target:** 가능한 와일드카드가 있는 DN을 사용하여 항목을 지정

또한 다음 편의 옵션에서 해당 속성을 설정합니다.

- **type:** 개체 유형 (사용자, 그룹 등)은 하위 트리 와 대상 필터를 설정합니다.
- **memberOf:** 그룹의 멤버이며, 대상 필터를 설정합니다.
- **TargetGroup :** 특정 그룹을 수정할 수 있는 액세스 권한을 부여합니다(예: 그룹 멤버십을 관리할 수 있는 권한 부여)

**IdM** 권한을 사용하면 어떤 오브젝트에 액세스할 수 있는 사용자와 이러한 오브젝트의 속성을 제어할 수 있습니다. **IdM**을 사용하면 개별 속성을 허용 또는 차단하거나 사용자, 그룹 또는 **sudo**와 같은 특정 **IdM** 기능의 전체 가시성을 모든 익명 사용자, 인증된 모든 사용자 또는 특정 권한 있는 사용자 그룹에 변경할 수 있습니다.

예를 들어, 권한에 대한 이 접근 방식의 유연성은 사용자 또는 그룹의 액세스 권한을 이러한 사용자 또는 그룹에 대한 액세스만 제한하려는 관리자에게 유용합니다. 이러한 사용자 또는 그룹에 액세스해야 하며 다른 섹션을 완전히 숨겨야 합니다.



참고

권한은 다른 권한을 포함할 수 없습니다.

### 21.1.2. 기본 관리 권한

관리 권한은 기본적으로 **IdM**과 함께 제공되는 권한입니다. 이러한 권한은 사용자가 생성한 다른 권한처럼 작동하며 다음과 같은 차이점이 있습니다.

- 해당 항목을 삭제하거나 이름, 위치 및 대상 속성을 수정할 수 없습니다.
- 세 가지 속성 세트가 있습니다.
  - 기본 속성, 사용자가 **IdM**에서 관리하므로 수정할 수 없습니다.
  - 사용자가 추가한 추가 속성인 포함 특성
  - 사용자가 제거된 속성인 제외된 특성

관리 권한은 **default** 및 **included** 특성 세트에 표시되지만 제외 세트에는 표시되지 않는 모든 특성에 적용됩니다.



참고

관리 권한을 삭제할 수는 없지만 해당 **bind** 유형을 권한으로 설정하고 모든 권한에서 관리 권한을 제거하면 이를 효과적으로 비활성화합니다.

모든 관리 권한의 이름은 **System:** (예: **System: Add Sudo rule** or **System: Modify Services**)로 시작합니다. 이전 버전의 IdM에서는 기본 권한에 다른 스키마를 사용했습니다. 예를 들어 사용자는 해당 항목을 삭제할 수 없으며 해당 사용자를 권한에만 할당할 수 있었습니다. 이러한 기본 권한의 대부분은 관리 되는 권한으로 전환되었지만 다음 권한에서는 여전히 이전 스키마를 사용합니다.

- 멤버십 자동 작성 작업 추가
- 설정 하위 항목 추가
- 복제 계약 추가
- 인증서 제거 **Hold**
- CA에서 인증서 상태 가져오기
- DNA 범위 읽기
- DNA 범위 수정
- PassSync Managers 설정 읽기
- PassSync Managers 설정 수정
- 복제 계약 읽기
- 복제 계약 수정
- 복제 계약 제거

- **LDBM 데이터베이스 구성 읽기**
- 요청 인증서
- **CA ACL을 무시하는 인증서 요청**
- 다른 호스트의 인증서 요청
- **CA에서 인증서 검색**
- 인증서 해지
- **IPA 설정 쓰기**



#### 참고

명령줄에서 관리 권한을 수정하려고 하면 시스템에서 수정할 수 없는 속성을 변경할 수 없으므로 명령이 실패합니다. 웹 UI에서 관리되는 권한을 수정하려고 하면 수정할 수 없는 특성이 비활성화됩니다.

### 21.1.3. IdM의 권한

권한은 역할에 적용 가능한 권한 그룹입니다.

권한은 단일 작업을 수행할 수 있는 권한을 제공하지만 성공하려면 여러 권한이 필요한 특정 IdM 작업이 있습니다. 따라서 권한은 특정 작업을 수행하는 데 필요한 다양한 권한을 결합합니다.

예를 들어 새 IdM 사용자에게 대한 계정을 설정하려면 다음 권한이 필요합니다.

- 새 사용자 항목 만들기
- 사용자 암호 재설정
- 기본 IPA 사용자 그룹에 새 사용자 추가

이러한 세 가지 하위 수준 작업을 라는 사용자 지정 권한 형식의 상위 수준 작업으로 결합하면 시스템 관리자가 역할을 보다 쉽게 관리할 수 있습니다. **IdM**에는 이미 여러 기본 권한이 포함되어 있습니다. 사용자 및 사용자 그룹 외에도 호스트 및 호스트 그룹과 네트워크 서비스에 권한이 할당됩니다. 이 방법을 사용하면 특정 네트워크 서비스를 사용하는 호스트 집합에서 일련의 사용자별로 작업을 세부적으로 제어할 수 있습니다.



#### 참고

권한에는 다른 권한이 포함되지 않을 수 있습니다.

### 21.1.4. IdM의 역할

역할은 역할에 지정된 사용자가 보유한 권한 목록입니다. 실제로 권한은 낮은 수준의 작업(사용자 항목 생성, 그룹에 항목을 추가 등)을 수행할 수 있는 기능을 부여합니다. 권한은 상위 수준의 작업(예: 지정된 그룹에서 새 사용자 생성)에 필요한 이러한 권한 중 하나 이상을 결합합니다. 역할은 필요에 따라 권한을 함께 수집합니다. 예를 들어 사용자 관리자 역할은 사용자를 추가, 수정, 삭제할 수 있습니다.



#### 중요

역할은 허용된 작업을 분류하는 데 사용됩니다. 권한 분리를 구현하거나 권한 에스컬레이션으로부터 보호하는 도구로는 사용되지 않습니다.



#### 참고

역할에 다른 역할을 포함할 수 없습니다.

### 21.1.5. ID 관리에서 사전 정의된 역할

**Red Hat Identity Management**는 다음과 같은 다양한 사전 정의 역할을 제공합니다.

표 21.1. ID 관리에서 사전 정의된 역할

| Role     | 권한                         | 설명                  |
|----------|----------------------------|---------------------|
| 등록 관리자   | 호스트 등록                     | 클라이언트 또는 호스트 등록 담당자 |
| helpdesk | 사용자 수정 및 암호 재설정, 그룹 멤버십 수정 | 간단한 사용자 관리 작업 수행    |

| Role      | 권한                                       | 설명                                 |
|-----------|------------------------------------------|------------------------------------|
| IT 보안 전문가 | Netgroups 관리자, HBAC 관리자, Sudo 관리자        | 호스트 기반 액세스 제어와 같은 보안 정책 관리 sudo 규칙 |
| IT 전문가    | 호스트 관리자, 호스트 그룹 관리자, 서비스 관리자, 자동 마운트 관리자 | 호스트 관리                             |
| 보안 아키텍트   | 위임 관리자, 복제 관리자, 쓰기 IPA 구성, 암호 정책 관리자     | ID 관리 환경 관리, 신뢰 생성, 복제 계약 생성       |
| 사용자 관리자   | 사용자 관리자, 그룹 관리자, 사용자 관리자 단계              | 사용자 및 그룹 생성                        |

## 21.2. IDM 웹 UI에서 권한 관리

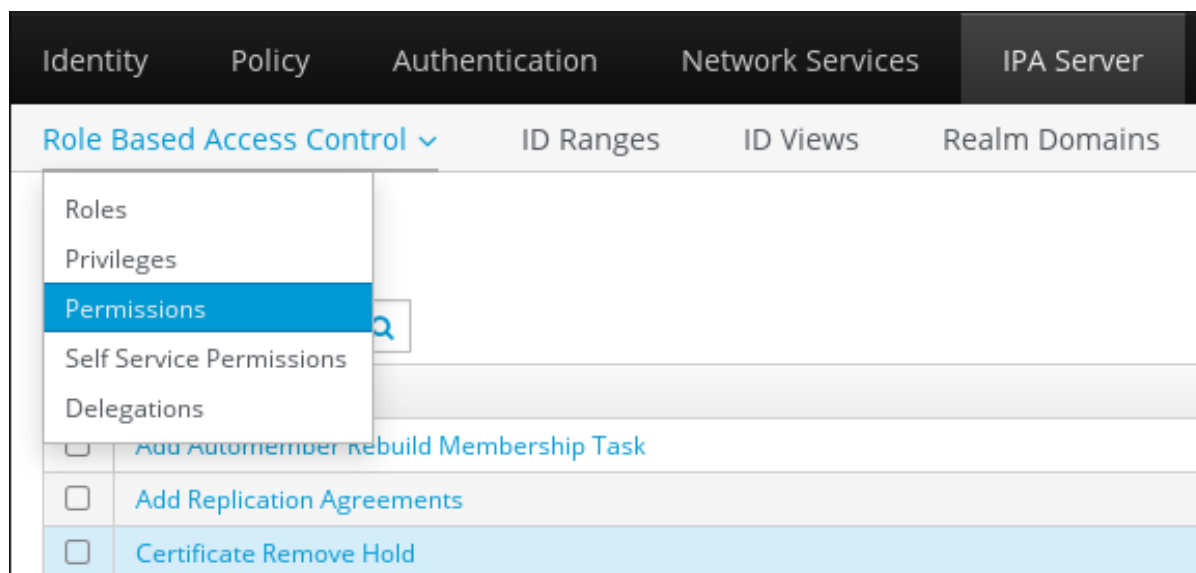
이 섹션에서는 웹 인터페이스(IdM Web UI)를 사용하여 IdM(Identity Management)에서 권한을 관리하는 방법을 설명합니다.

### 사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 [IdM 웹 UI 액세스](#)를 참조하십시오.

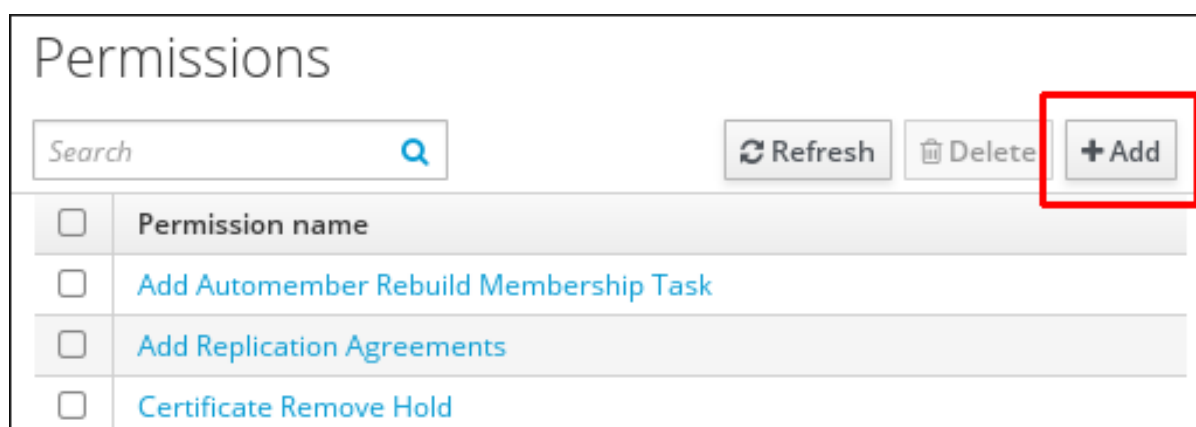
### 절차

1. 새 권한을 추가하려면 IPA 서버 탭에서 **Role-Based Access Control** (역할 기반 액세스 제어) 하위 메뉴를 열고 권한을 선택합니다.



2.

권한 목록이 열립니다. 권한 목록 상단에서 **Add** 버튼을 클릭합니다.



3.

권한 추가 양식이 열립니다. 새 권한의 이름을 지정하고 그에 따라 해당 속성을 정의합니다.

Add Permission

Permission name \*

New Permission

Bind rule type

☒ permission
☐ all
☐ anonymous

Granted rights \*

☒ read
☐ search
☐ compare
  
☐ write
☐ add
☐ delete
  
☐ all

Type

Subtree \*

dc=demo1,dc=freeipa,dc=org

Extra target filter

Add

Target DN

Member of group

employees

Undo

Add

Effective attributes

uid

Undo

loginshell

Undo

Add

\* Required field

Add

Add and Add Another

Add and Edit

Cancel

4.

적절한 바인딩 규칙 유형을 선택합니다.

•

권한은 기본 권한 유형이며 권한 및 역할을 통해 액세스 권한을 부여합니다.

- **all** 는 인증된 모든 사용자에게 권한이 적용되도록 지정합니다.
- **anonymous** 는 인증되지 않은 사용자를 포함하여 모든 사용자에게 권한이 적용되도록 지정합니다.



## 참고

기본이 아닌 바인딩 규칙 유형의 권한을 권한에 추가할 수 없습니다. 또한 기본이 아닌 바인딩 규칙 유형에 이미 있는 권한도 설정할 수 없습니다.

5. 부여된 권한에서 이 권한을 부여할 권한을 선택합니다.
  6. 권한에 대한 대상 항목을 식별하는 메서드를 정의합니다.
- **type**은 **user**, **host** 또는 **service**와 같은 항목 유형을 지정합니다. **Type** 설정에 대한 값을 선택하면 해당 항목 유형에 대해 이 **ACI**를 통해 액세스할 수 있는 모든 속성 목록이 **Effective Attributes** 아래에 표시됩니다. **Type** 을 정의하면 **Subtree** 및 **Target DN** 을 사전 정의된 값 중 하나로 설정합니다.
  - 하위 트리 (필수)는 하위 트리를 지정합니다. 이 하위 트리 항목 아래의 모든 항목을 대상으로 합니다. **Subtree** 에서 와일드카드 또는 존재하지 않는 도메인 이름(**DN**)을 허용하지 않으므로 기존 하위 트리 항목을 제공합니다. 예: **cn=automount,dc=example,dc=com**
  - 추가 대상 필터 는 **LDAP** 필터를 사용하여 권한이 적용되는 항목을 식별합니다. 필터는 유효한 **LDAP** 필터(예: **!(objectclass=posixgroup)**) **IdM**에서 지정된 필터의 유효성을 자동으로 확인할 수 있습니다. 유효하지 않은 필터를 입력하면 **IdM**에서 권한을 저장하려고 할 때 이에 대해 경고합니다.
  - 대상 **DN** 은 도메인 이름(**DN**)을 지정하고 와일드카드를 허용합니다. 예: **uid=\*,cn=users,cn=accounts,dc=com**
  - 그룹의 멤버는 대상 필터를 지정된 그룹의 멤버로 설정합니다. 필터 설정을 지정하고 추가를 클릭하면 **IdM**에서 필터를 검증합니다. 모든 권한 설정이 올바르면 **IdM**에서 검색을 수행합니다. 일부 권한 설정이 올바르지 않으면 **IdM**에서 어떤 설정이 잘못 설정되었는지 알려주는 메시지를 표시합니다.

7.

권한에 속성을 추가합니다.

- 유형 을 설정하는 경우 사용 가능한 **ACI** 속성 목록에서 효과를 적용할 수 있는 속성을 선택합니다.
- **Type** 을 사용하지 않은 경우 **Effective attributes** 필드에 작성하여 속성을 수동으로 추가합니다. 한 번에 단일 속성을 추가합니다. 여러 속성을 추가하려면 **Add** 를 클릭하여 다른 입력 필드를 추가합니다.



중요

권한의 속성을 설정하지 않으면 권한에는 기본적으로 모든 속성이 포함됩니다.

8.

양식 하단에 있는 **Add** 버튼을 사용하여 권한을 추가합니다.

- 추가 버튼을 클릭하여 권한을 저장하고 권한 목록으로 돌아갑니다.
- 또는 권한을 저장하고 **Add and Add another** 버튼을 클릭하여 동일한 양식에 추가 권한을 계속 추가할 수 있습니다.
- **Add** 및 **Edit** 버튼을 사용하면 새로 만든 권한을 저장하고 계속 편집할 수 있습니다.

9.

**선택 사항:** 권한 목록에서 해당 이름을 클릭하여 권한 설정 페이지를 표시하여 기존 권한의 속성을 편집할 수도 있습니다.

10.

**선택 사항:** 기존 권한을 제거해야 하는 경우 목록에서 이름 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭하여 권한 제거 대화 상자를 표시합니다.



## 참고

수정할 수 없는 특성은 **IdM 웹 UI**에서 비활성화되어 있으며 관리 권한을 완전히 삭제할 수 없습니다.

그러나 모든 권한에서 관리되는 권한을 제거하여 바인딩 유형이 권한으로 설정된 관리되는 권한을 효과적으로 비활성화할 수 있습니다.

예를 들어, 권한을 가진 사용자가 **engineers** 그룹에 **member** 특성을 쓰도록 하려면 (관리자는 멤버를 추가 또는 제거할 수 있음)

Add permission

Permission name \*

Bind rule type
☒ permission
☐ all
☐ anonymous

Granted rights \*

☐ read
☐ search
☐ compare

☒ write
☐ add
☐ delete

☐ all

Type

Subtree \*

Extra target filter

Target DN

Member of group

Effective attributes

\* Required field

Add
Add and Add Another
Add and Edit
Cancel

### 21.3. IDM WEBUI에서 권한 관리

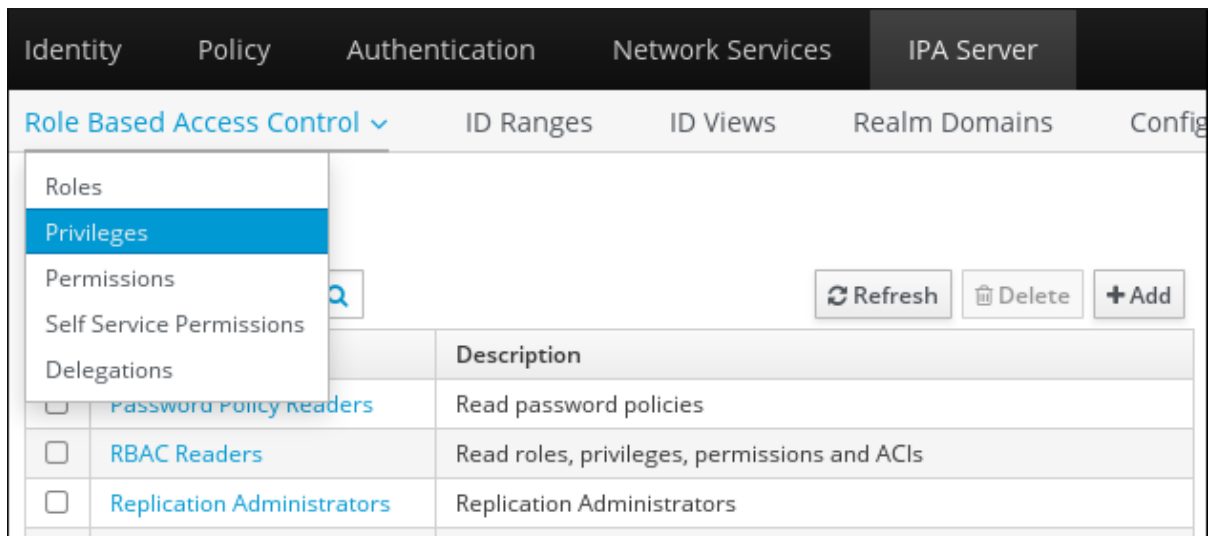
이 섹션에서는 웹 인터페이스(IdM Web UI)를 사용하여 IdM의 권한을 관리하는 방법을 설명합니다.

사전 요구 사항

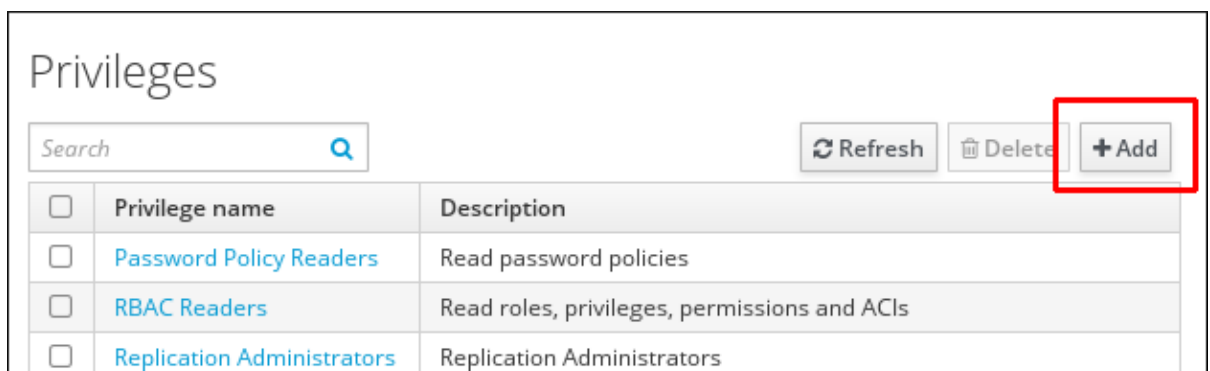
- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.**
- **기존 권한. 권한에 대한 자세한 내용은 IdM 웹 UI에서 권한 관리를 참조하십시오.**

## 절차

1. 새 권한을 추가하려면 **IPA** 서버 탭에서 **Role-Based Access Control** 하위 메뉴를 열고 **Privileges** 를 선택합니다.

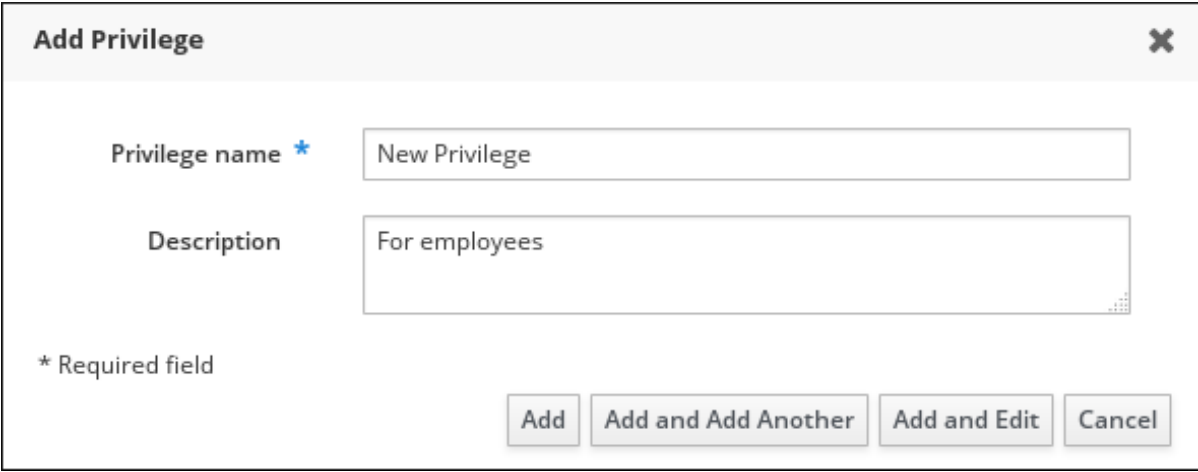


2. 권한 목록이 열립니다. 권한 목록 상단에서 **Add** 버튼을 클릭합니다.



3.

권한 추가 양식이 열립니다. 권한 이름 및 설명을 입력합니다.



The image shows a modal dialog box titled "Add Privilege" with a close button (X) in the top right corner. Inside the dialog, there are two input fields. The first is labeled "Privilege name" with a blue asterisk indicating it is required; it contains the text "New Privilege". The second is labeled "Description" and contains the text "For employees". Below the input fields, there is a note "\* Required field". At the bottom right of the dialog, there are four buttons: "Add", "Add and Add Another", "Add and Edit", and "Cancel".

4.

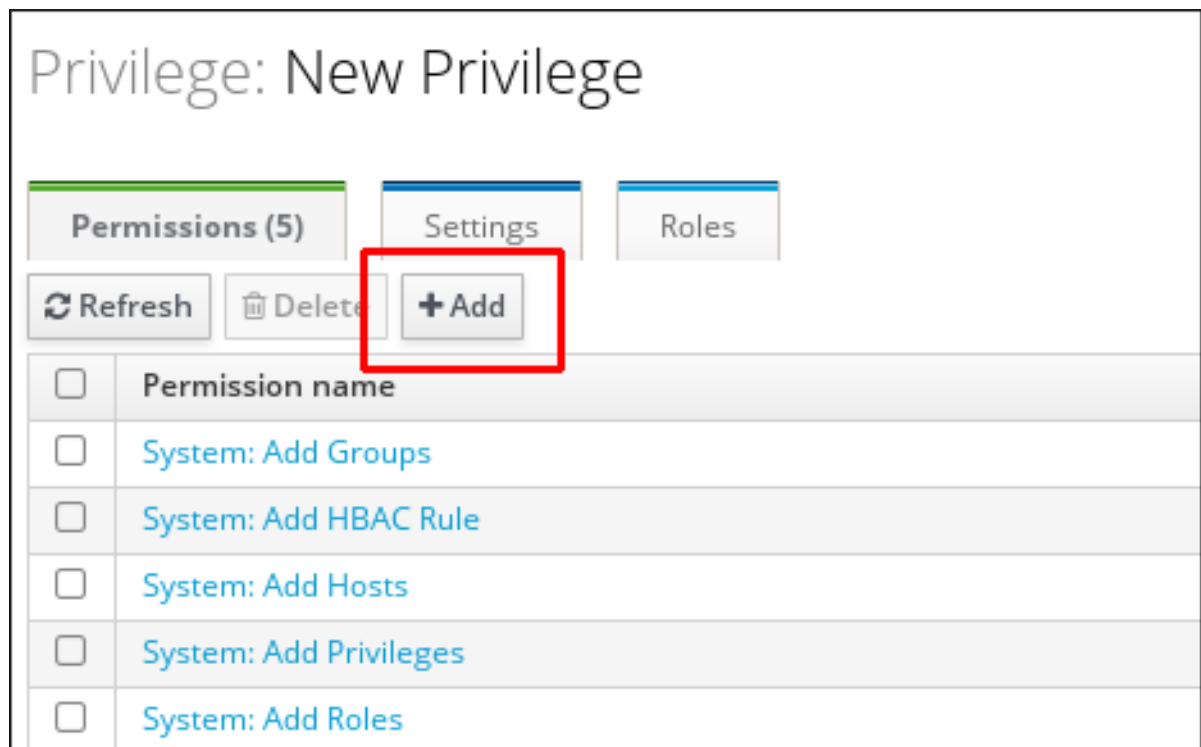
새 권한을 저장하려면 **Add and Edit** (추가 및 편집) 버튼을 클릭하여 새 권한을 저장하고 권한 구성 페이지로 계속 이동하여 권한을 추가합니다.

5.

권한 목록에서 권한 이름을 클릭하여 권한 속성을 편집합니다. 권한 구성 페이지가 열립니다.

6.

**Permissions** 탭에는 선택한 권한에 포함된 권한 목록이 표시됩니다. 목록 상단에 있는 **Add** 버튼을 클릭하여 권한에 권한을 추가합니다.



7.

추가할 각 권한의 이름 옆에 있는 확인란을 선택하고 > 버튼을 사용하여 **Prospective** 열로 권한을 이동합니다.

Add Privilege New Privilege into Permissions

Filter available Permissions

Filter

Available

|                          |                                        |
|--------------------------|----------------------------------------|
|                          | Policy costemplate                     |
| <input type="checkbox"/> | System: Add HBAC Service Groups        |
| <input type="checkbox"/> | System: Add HBAC Services              |
| <input type="checkbox"/> | System: Add Hostgroups                 |
| <input type="checkbox"/> | System: Add krbPrincipalName to a Host |
| <input type="checkbox"/> | System: Add Netgroups                  |
| <input type="checkbox"/> | System: Add SELinux User Maps          |
| <input type="checkbox"/> | System: Add Sudo Command Group         |
| <input type="checkbox"/> | System: Delete Group Password Policy   |

>
<

Prospective

|                          |                          |
|--------------------------|--------------------------|
| <input type="checkbox"/> | Permission name          |
| <input type="checkbox"/> | System: Add Services     |
| <input type="checkbox"/> | System: Add Sudo Command |

Add

Cancel

8.

**Add (추가)** 버튼을 클릭하여 확인합니다.

9.

**선택 사항:** 권한을 제거해야 하는 경우 관련 권한 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭합니다. 권한에서 권한 제거 대화 상자가 열립니다.

10.

**선택 사항:** 기존 권한을 삭제해야 하는 경우 목록에서 이름 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭합니다. 권한 제거 대화 상자가 열립니다.

## 21.4. IDM 웹 UI에서 역할 관리

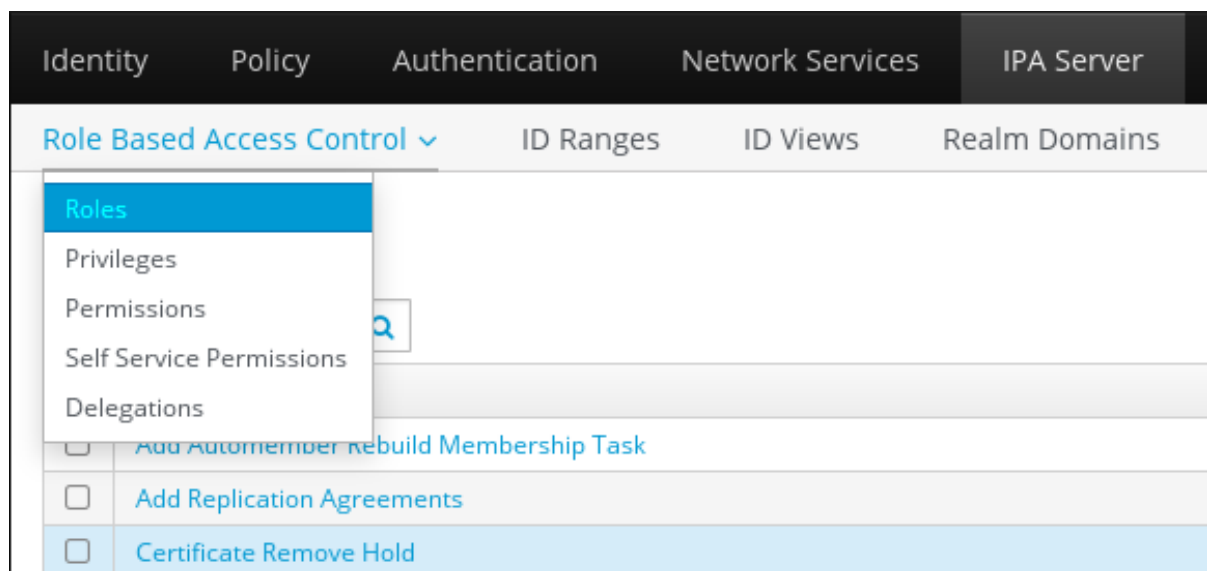
이 섹션에서는 웹 인터페이스(IdM Web UI)를 사용하여 IdM(Identity Management)에서 역할을 관리하는 방법을 설명합니다.

사전 요구 사항

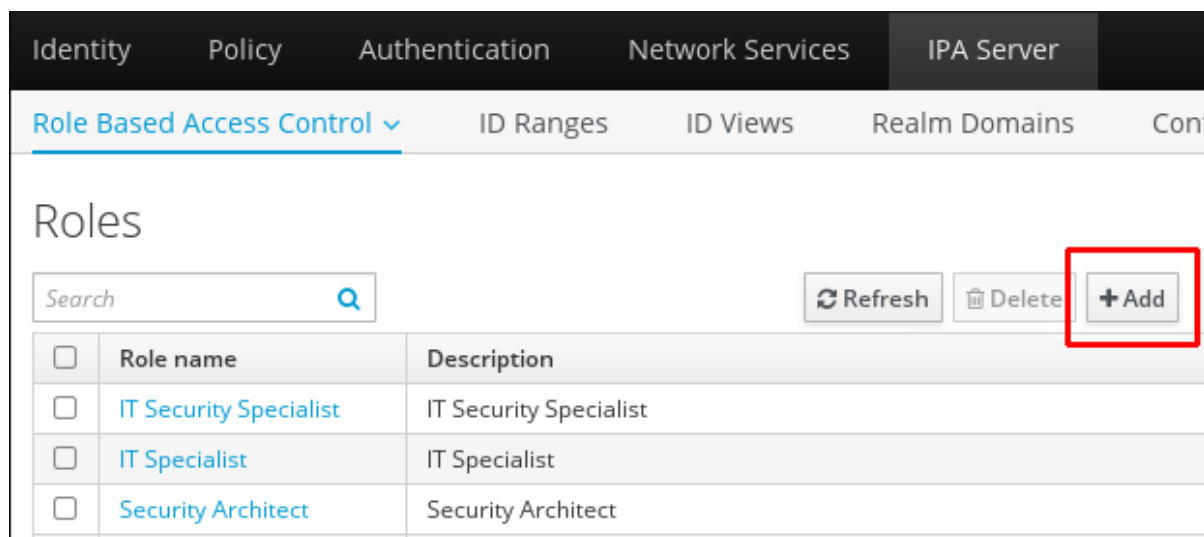
- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **IdM 웹 UI**에 로그인되어 있습니다. 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.
- 기존 권한. 권한에 대한 자세한 내용은 [IdM 웹 UI의 권한](#) 관리를 참조하십시오.

## 절차

1. 새 역할을 추가하려면 **IPA** 서버 탭에서 **Role-Based Access Control** 하위 메뉴를 열고 **Roles**를 선택합니다.



2. 역할 목록이 열립니다. 역할 기반 액세스 제어 지침 목록 상단에 있는 추가 버튼을 클릭합니다.

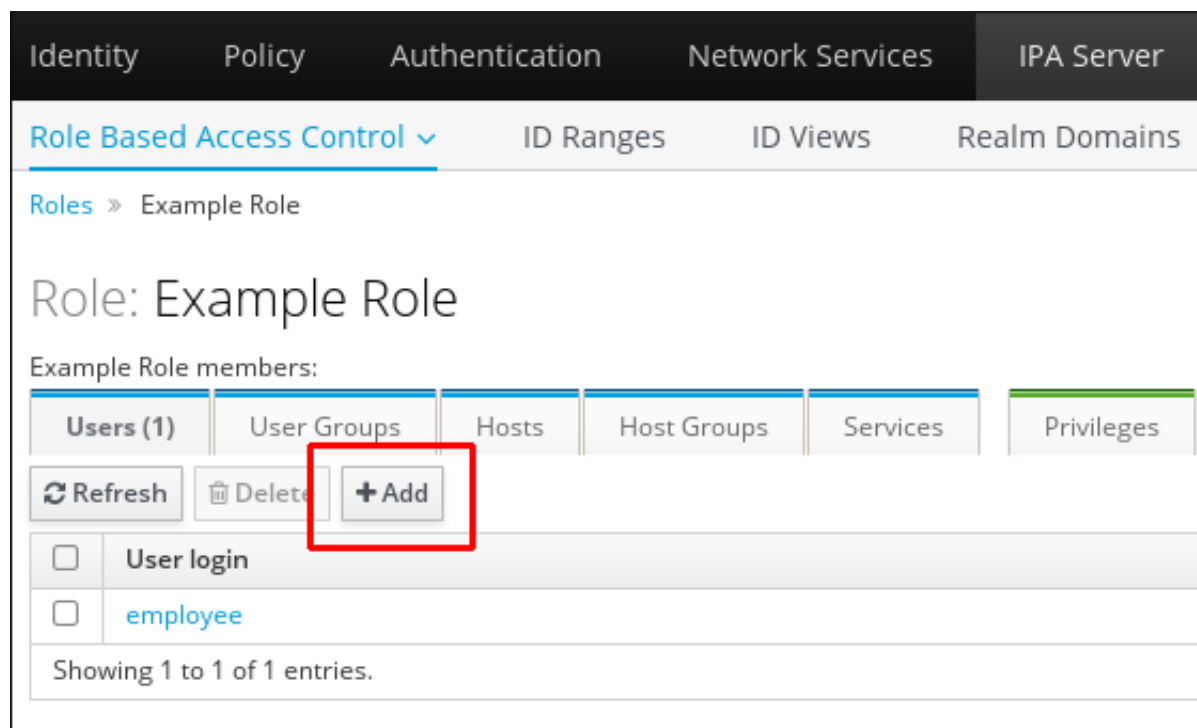


3. 역할 추가 양식이 열립니다. 역할 이름 및 설명을 입력합니다.

The 'Add Role' dialog box is shown with the following fields and buttons:

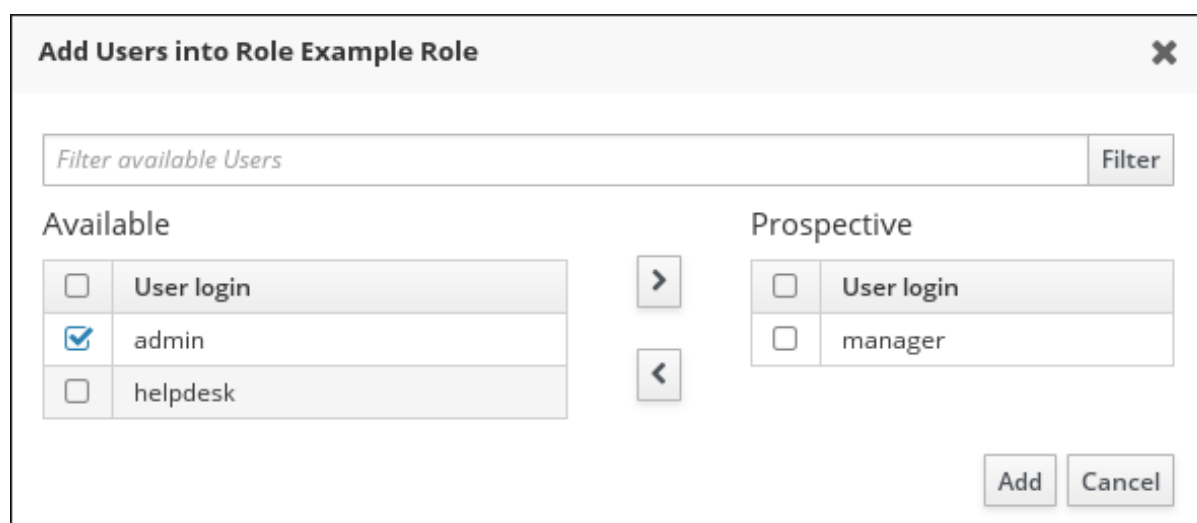
- Role name \***: Text input field containing 'Example Role'.
- Description**: Text area containing 'For engineers'.
- \* Required field**: Label indicating required fields.
- Buttons**: 'Add', 'Add and Add Another', 'Add and Edit', and 'Cancel'.

4. 추가 및 편집 버튼을 클릭하여 새 역할을 저장하고 역할 구성 페이지로 이동하여 권한 및 사용자를 추가합니다.
5. 역할 목록에서 역할 이름을 클릭하여 역할 속성을 편집합니다. 역할 구성 페이지가 열립니다.
6. 관련 목록 상단에서 **Add(추가)** 버튼을 클릭하여 **Users, Users Groups, Hosts, Host Groups or Services** 탭을 사용하여 멤버를 추가합니다.



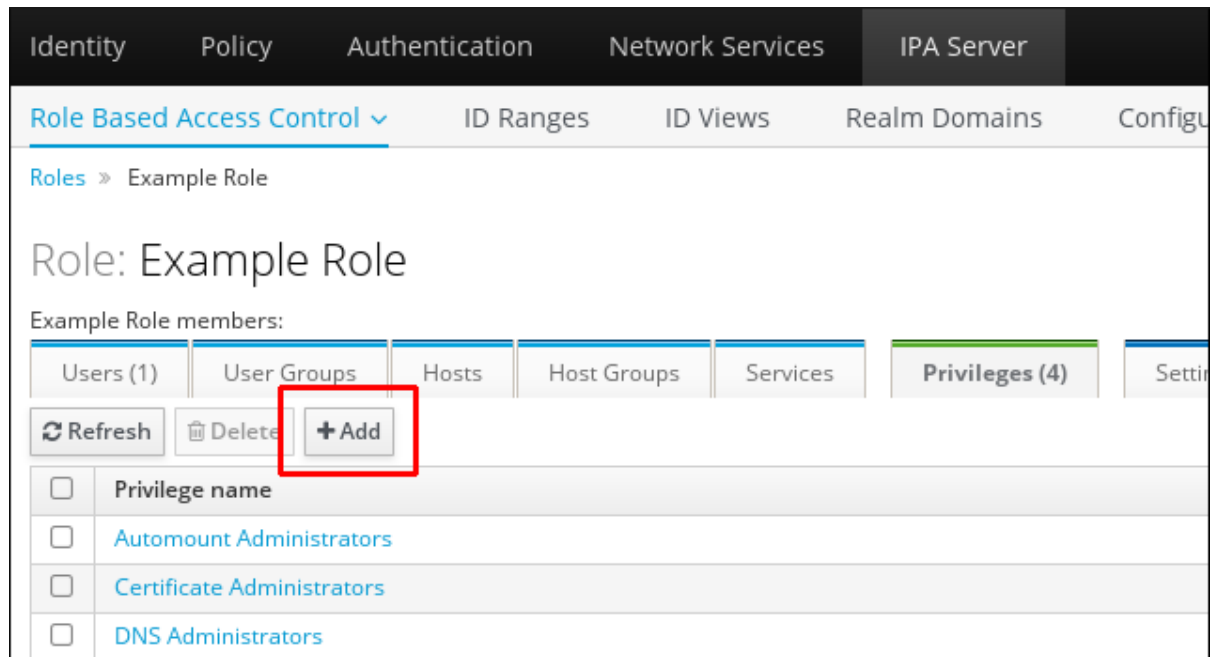
7.

창이 열리면 왼쪽의 멤버를 선택하고 > 버튼을 사용하여 **Prospective** 열로 이동합니다.



8.

**Privileges** (권한) 탭 상단에서 추가 를 클릭합니다.



9.

왼쪽의 권한을 선택하고 **>** 버튼을 사용하여 **Prospective** 열로 이동합니다.

Add Role Example Role into Privileges

Filter available Privileges
Filter

Available

|                          |                                  |
|--------------------------|----------------------------------|
| <input type="checkbox"/> | Privilege name                   |
| <input type="checkbox"/> | ADTrust Agents                   |
| <input type="checkbox"/> | Automember Readers               |
| <input type="checkbox"/> | Automember Task Administrator    |
| <input type="checkbox"/> | DNS Servers                      |
| <input type="checkbox"/> | HBAC Administrator               |
| <input type="checkbox"/> | Host Enrollment                  |
| <input type="checkbox"/> | IPA Masters Readers              |
| <input type="checkbox"/> | Kerberos Ticket Policy Readers   |
| <input type="checkbox"/> | Modify Group membership          |
| <input type="checkbox"/> | Modify Users and Reset passwords |
| <input type="checkbox"/> | Netgroups Administrators         |
| <input type="checkbox"/> | New Privilege                    |
| <input type="checkbox"/> | Password Policy Administrator    |
| <input type="checkbox"/> | Password Policy Readers          |

Prospective

|                          |                           |
|--------------------------|---------------------------|
| <input type="checkbox"/> | Privilege name            |
| <input type="checkbox"/> | Group Administrators      |
| <input type="checkbox"/> | Host Administrators       |
| <input type="checkbox"/> | Host Group Administrators |

Add Cancel

10.

저장 버튼을 클릭합니다.

11.

**선택 사항:** 역할에서 권한 또는 멤버를 제거해야 하는 경우 제거할 엔티티 이름 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭합니다. 대화 상자가 열립니다.

12.

**선택 사항:** 기존 역할을 제거해야 하는 경우 목록에서 이름 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭하여 역할 제거 대화 상자를 표시합니다.

## 22장. ANSIBLE 플레이북을 사용하여 IDM을 관리하기 위한 환경 준비

Red Hat Ansible Engine으로 작업할 때 IdM(Identity Management)을 관리하는 시스템 관리자는 다음을 수행하는 것이 좋습니다.

- 홈 디렉터리에서 **Ansible** 플레이북 전용 하위 디렉터리를 만듭니다(예: `~/MyPlaybooks` ).
- `/usr/share/doc/ansible-freeipa/*` 및 `/usr/share/doc/rhel-system-roles/*` 디렉터리 및 하위 디렉터리에서 샘플 **Ansible** 플레이북을 `~/MyPlaybooks` 디렉터리에 복사 및 조정합니다.
- `~/MyPlaybooks` 디렉터리에 인벤토리 파일을 포함합니다.

이 방법을 사용하면 모든 플레이북을 한 곳에서 찾을 수 있으며 **root** 권한을 호출하지 않고도 플레이북을 실행할 수 있습니다.



### 참고

`ipaserver`, `ipareplica`, `ipaclient` 및 `ipabackup` `ansible-freeipa` 역할을 실행하려면 관리형 노드에서 **root** 권한만 있으면 됩니다. 이러한 역할을 수행하려면 디렉터리 및 `dnf` 소프트웨어 패키지 관리자에 대한 권한 있는 액세스 권한이 필요합니다.

이 섹션에서는 `~/MyPlaybooks` 디렉터리를 생성하고 이를 사용하여 **Ansible** 플레이북을 저장하고 실행하는 방법을 설명합니다.

### 사전 요구 사항

- 관리 노드, `server.idm.example.com` 및 `replica.idm.example.com` 에 IdM 서버를 설치했습니다.
- 제어 노드에서 직접 관리 노드 `server.idm.example.com` 및 `replica.idm.example.com` 에 로그인할 수 있도록 **DNS** 및 네트워킹을 구성했습니다.
- IdM 관리자 암호를 알고 있습니다.

### 절차

1. 홈 디렉터리에 **Ansible** 구성 및 플레이북의 디렉터리를 생성합니다.

```
$ mkdir ~/MyPlaybooks/
```

2. ~/MyPlaybooks/ 디렉터리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3. 다음 콘텐츠를 사용하여 ~/MyPlaybooks/ansible.cfg 파일을 생성합니다.

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

[privilege_escalation]
become=True
```

4. 다음 콘텐츠를 사용하여 ~/MyPlaybooks/inventory 파일을 만듭니다.

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

이 구성은 해당 위치에 있는 호스트에 대해 **eu** 및 **us**의 두 개의 호스트 그룹을 정의합니다. 또한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트가 포함된 **ipaserver** 호스트 그룹을 정의합니다.

5. [선택 사항] SSH 공개 및 개인 키를 생성합니다. 테스트 환경에서 액세스를 단순화하려면 개인 키에 암호를 설정하지 마십시오.

```
$ ssh-keygen
```

6. SSH 공개 키를 각 관리 노드의 IdM 관리자 계정에 복사합니다.

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

이러한 명령을 실행하려면 **IdM** 관리자 암호를 입력해야 합니다.

#### 추가 리소스

- [Ansible 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오.](#)
- [인벤토리 빌드 방법을 참조하십시오.](#)

## 23장. ANSIBLE 플레이북을 사용하여 IDM에서 역할 기반 액세스 제어 관리

**RBAC(역할 기반 액세스 제어)**는 역할 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. **IdM(Identity Management)**의 **RBAC** 구성 요소는 역할, 권한 및 권한입니다.

- 권한은 사용자 추가 또는 삭제, 그룹 수정, 읽기 액세스 등의 특정 작업을 수행할 수 있는 권한을 부여합니다.
- 권한은 예를 들어 새 사용자를 추가하는 데 필요한 모든 권한을 결합합니다.
- 역할에 따라 사용자, 사용자 그룹, 호스트 또는 호스트 그룹에 일련의 권한이 부여됩니다.

특히 대규모 회사에서 **RBAC**를 사용하면 개별 책임 영역이 있는 관리자의 계층적 시스템을 만드는 데 도움이 될 수 있습니다.

이 장에서는 **Ansible** 플레이북을 사용하여 **RBAC**를 관리할 때 수행되는 다음 작업을 설명합니다.

- **IdM의 권한**
- **기본 관리 권한**
- **IdM의 권한**
- **IdM의 역할**
- **IdM에서 사전 정의된 역할**
- **Ansible을 사용하여 권한이 있는 IdM RBAC 역할이 있는지 확인**
- **Ansible을 사용하여 IdM RBAC 역할이 없는지 확인**

- **Ansible을 사용하여 IdM RBAC 역할에 사용자 그룹이 할당되었는지 확인**
- **Ansible을 사용하여 특정 사용자가 IdM RBAC 역할에 할당되지 않았는지 확인**
- **Ansible을 사용하여 서비스가 IdM RBAC 역할의 멤버인지 확인**
- **Ansible을 사용하여 호스트가 IdM RBAC 역할의 멤버인지 확인**
- **Ansible을 사용하여 호스트 그룹이 IdM RBAC 역할의 멤버인지 확인**

### 23.1. IDM의 권한

권한은 역할 기반 액세스 제어의 가장 낮은 수준 단위이며 해당 작업이 적용되는 **LDAP** 항목과 함께 작업을 정의합니다. 빌딩 블록에 비해 필요한 만큼의 권한에 권한을 할당할 수 있습니다. 하나 이상의 권한은 허용되는 작업을 정의합니다.

- **write**
- **read**
- **search**
- **비교**
- **add**
- **delete**
- **all**

이러한 작업은 다음 세 가지 기본 대상에 적용됩니다.

- **subtree:** 도메인 이름(DN); 이 DN 아래의 하위 트리
- **대상 필터:** LDAP 필터
- **target:** 가능한 와일드카드가 있는 DN을 사용하여 항목을 지정

또한 다음 편의 옵션에서 해당 속성을 설정합니다.

- **type:** 개체 유형 (사용자, 그룹 등)은 하위 트리와 대상 필터를 설정합니다.
- **memberOf:** 그룹의 멤버이며, 대상 필터를 설정합니다.
- **TargetGroup :** 특정 그룹을 수정할 수 있는 액세스 권한을 부여합니다(예: 그룹 멤버십을 관리할 수 있는 권한 부여)

IdM 권한을 사용하면 어떤 오브젝트에 액세스할 수 있는 사용자와 이러한 오브젝트의 속성을 제어할 수 있습니다. IdM을 사용하면 개별 속성을 허용 또는 차단하거나 사용자, 그룹 또는 **sudo**와 같은 특정 IdM 기능의 전체 가시성을 모든 익명 사용자, 인증된 모든 사용자 또는 특정 권한 있는 사용자 그룹에 변경할 수 있습니다.

예를 들어, 권한에 대한 이 접근 방식의 유연성은 사용자 또는 그룹의 액세스 권한을 이러한 사용자 또는 그룹에 대한 액세스만 제한하려는 관리자에게 유용합니다. 이러한 사용자 또는 그룹에 액세스해야 하며 다른 섹션을 완전히 숨겨야 합니다.



참고

권한은 다른 권한을 포함할 수 없습니다.

## 23.2. 기본 관리 권한

관리 권한은 기본적으로 IdM과 함께 제공되는 권한입니다. 이러한 권한은 사용자가 생성한 다른 권한처럼 작동하며 다음과 같은 차이점이 있습니다.

- 해당 항목을 삭제하거나 이름, 위치 및 대상 속성을 수정할 수 없습니다.
- 세 가지 속성 세트가 있습니다.
  - 기본 속성, 사용자가 **IdM**에서 관리하므로 수정할 수 없습니다.
  - 사용자가 추가한 추가 속성인 포함 특성
  - 사용자가 제거된 속성인 제외된 특성

관리 권한은 **default** 및 **included** 특성 세트에 표시되지만 제외 세트에는 표시되지 않는 모든 특성에 적용됩니다.



#### 참고

관리 권한을 삭제할 수는 없지만 해당 **bind** 유형을 권한으로 설정하고 모든 권한에서 관리 권한을 제거하면 이를 효과적으로 비활성화합니다.

모든 관리 권한의 이름은 **System:** (예: **System: Add Sudo rule** or **System: Modify Services**)로 시작합니다. 이전 버전의 **IdM**에서는 기본 권한에 다른 스키마를 사용했습니다. 예를 들어 사용자는 해당 항목을 삭제할 수 없으며 해당 사용자를 권한에만 할당할 수 있었습니다. 이러한 기본 권한의 대부분은 관리 되는 권한으로 전환되었지만 다음 권한에서는 여전히 이전 스키마를 사용합니다.

- 멤버십 자동 작성 작업 추가
- 설정 하위 항목 추가
- 복제 계약 추가
- 인증서 제거 **Hold**

- **CA에서 인증서 상태 가져오기**
- **DNA 범위 읽기**
- **DNA 범위 수정**
- **PassSync Managers 설정 읽기**
- **PassSync Managers 설정 수정**
- 복제 계약 읽기
- 복제 계약 수정
- 복제 계약 제거
- **LDBM 데이터베이스 구성 읽기**
- 요청 인증서
- **CA ACL을 무시하는 인증서 요청**
- 다른 호스트의 인증서 요청
- **CA에서 인증서 검색**
- 인증서 해지

•

## IPA 설정 쓰기



### 참고

명령줄에서 관리 권한을 수정하려고 하면 시스템에서 수정할 수 없는 속성을 변경할 수 없으므로 명령이 실패합니다. 웹 UI에서 관리되는 권한을 수정하려고 하면 수정할 수 없는 특성이 비활성화됩니다.

## 23.3. IDM의 권한

권한은 역할에 적용 가능한 권한 그룹입니다.

권한은 단일 작업을 수행할 수 있는 권한을 제공하지만 성공하려면 여러 권한이 필요한 특정 **IdM** 작업이 있습니다. 따라서 권한은 특정 작업을 수행하는 데 필요한 다양한 권한을 결합합니다.

예를 들어 새 **IdM** 사용자에게 대한 계정을 설정하려면 다음 권한이 필요합니다.

•

새 사용자 항목 만들기

•

사용자 암호 재설정

•

기본 **IPA** 사용자 그룹에 새 사용자 추가

이러한 세 가지 하위 수준 작업을 라는 사용자 지정 권한 형식의 상위 수준 작업으로 결합하면 시스템 관리자가 역할을 보다 쉽게 관리할 수 있습니다. **IdM**에는 이미 여러 기본 권한이 포함되어 있습니다. 사용자 및 사용자 그룹 외에도 호스트 및 호스트 그룹과 네트워크 서비스에 권한이 할당됩니다. 이 방법을 사용하면 특정 네트워크 서비스를 사용하는 호스트 집합에서 일련의 사용자별로 작업을 세부적으로 제어할 수 있습니다.



### 참고

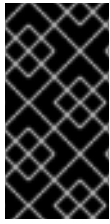
권한에는 다른 권한이 포함되지 않을 수 있습니다.

## 23.4. IDM의 역할

역할은 역할에 지정된 사용자가 보유한 권한 목록입니다.

실제로 권한은 낮은 수준의 작업(사용자 항목 생성, 그룹에 항목을 추가 등)을 수행할 수 있는 기능을 부여합니다. 권한은 상위 수준의 작업(예: 지정된 그룹에서 새 사용자 생성)에 필요한 이러한 권한 중 하나

이상을 결합합니다. 역할은 필요에 따라 권한을 함께 수집합니다. 예를 들어 사용자 관리자 역할은 사용자를 추가, 수정, 삭제할 수 있습니다.



#### 중요

역할은 허용된 작업을 분류하는 데 사용됩니다. 권한 분리를 구현하거나 권한 에스컬레이션으로부터 보호하는 도구로는 사용되지 않습니다.



#### 참고

역할에 다른 역할을 포함할 수 없습니다.

### 23.5. ID 관리에서 사전 정의된 역할

Red Hat Identity Management는 다음과 같은 다양한 사전 정의된 역할을 제공합니다.

표 23.1. ID 관리에서 사전 정의된 역할

| Role      | 권한                                       | 설명                                 |
|-----------|------------------------------------------|------------------------------------|
| 등록 관리자    | 호스트 등록                                   | 클라이언트 또는 호스트 등록 담당자                |
| helpdesk  | 사용자 수정 및 암호 재설정, 그룹 멤버십 수정               | 간단한 사용자 관리 작업 수행                   |
| IT 보안 전문가 | Netgroups 관리자, HBAC 관리자, Sudo 관리자        | 호스트 기반 액세스 제어와 같은 보안 정책 관리 sudo 규칙 |
| IT 전문가    | 호스트 관리자, 호스트 그룹 관리자, 서비스 관리자, 자동 마운트 관리자 | 호스트 관리                             |
| 보안 아키텍트   | 위임 관리자, 복제 관리자, 쓰기 IPA 구성, 암호 정책 관리자     | ID 관리 환경 관리, 신뢰 생성, 복제 계약 생성       |
| 사용자 관리자   | 사용자 관리자, 그룹 관리자, 사용자 관리자 단계              | 사용자 및 그룹 생성                        |

### 23.6. ANSIBLE을 사용하여 권한이 있는 IDM RBAC 역할이 있는지 확인

기본 역할보다 IdM(Identity Management) 리소스에 대한 역할 기반 액세스(RBAC)를 보다 세밀하게 제어하려면 사용자 지정 역할을 생성합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 새 **IdM** 사용자 지정 역할에 대한 권한을 정의하고 해당 역할이 있는지 확인하는 방법을 설명합니다. 이 예제에서 새 **user\_and\_host\_administrator** 역할에는 기본적으로 **IdM**에 있는 다음 권한의 고유한 조합이 포함되어 있습니다.

- 그룹 관리자
- 사용자 관리자
- 사용자 관리자 단계
- 그룹 관리자

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
- **Ansible** 인벤토리 파일은 **~/ <MyPlaybooks>/** 디렉터리에 있습니다.

#### 절차

1. **~/ <MyPlaybooks> /** 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** 디렉터리에 있는 **role-member-user-present.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-user-present.yml role-member-user-present-copy.yml
```

- 3. 편집할 **role-member-user-present-copy.yml Ansible** 플레이북 파일을 엽니다.

- 4. **iparole** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 새 역할의 이름으로 설정합니다.
- 새 역할에 포함하려는 **IdM** 권한 이름으로 권한 목록을 설정합니다.
- 선택적으로 사용자 변수를 새 역할을 부여하려는 사용자 이름으로 설정합니다.
- 선택적으로 그룹 변수를 새 역할을 부여하려는 그룹 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: user_and_host_administrator
    user: idm_user01
    group: idm_group01
    privilege:
      - Group Administrators
      - User Administrators
      - Stage User Administrators
      - Group Administrators
```

- 5. 파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-user-present-copy.yml
```

추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-role** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리에서 샘플 플레이북을 참조하십시오.

### 23.7. ANSIBLE을 사용하여 IDM RBAC 역할이 없는지 확인

**IdM(Identity Management)**에서 **RBAC(역할 기반 액세스 제어)**를 관리하는 시스템 관리자는 관리자가 실수로 사용자에게 이를 할당하지 않도록 더 이상 사용되지 않도록 할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 역할이 없는지 확인하는 방법을 설명합니다. 아래 예제에서는 사용자 지정 **user\_and\_host\_administrator** 역할이 **IdM**에 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 [ansible-freeipa](#) 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 [Ansible 인벤토리 파일](#)을 생성했습니다.

- **Ansible** 인벤토리 파일은 `~/<MyPlaybooks>/` 디렉터리에 있습니다.

## 절차

1. `~/<MyPlaybooks>/` 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. `/usr/share/doc/ansible-freeipa/playbooks/role/` 디렉터리에 있는 `role-is-absent.yml` 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-is-absent.yml role-is-absent-copy.yml
```

3. 편집할 `role-is-absent-copy.yml` **Ansible** 플레이북 파일을 엽니다.
4. `iparole` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자의 암호로 설정합니다.
- `name` 변수를 역할의 이름으로 설정합니다.
- `state` 변수가 `absent` 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: user_and_host_administrator
    state: absent
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-is-absent-copy.yml
```

#### 추가 리소스

- [Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오.](#)
- [IdM의 역할을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리에서 샘플 플레이북을 참조하십시오.](#)

### 23.8. ANSIBLE을 사용하여 IDM RBAC 역할에 사용자 그룹이 할당되었는지 확인

**IdM(Identity Management)**에서 **RBAC(역할 기반 액세스 제어)**를 관리하는 시스템 관리자는 특정 사용자 그룹(예: **junior 관리자**)에 역할을 할당할 수 있습니다.

다음 예제에서는 **Ansible** 플레이북을 사용하여 기본 제공 **IdM RBAC** 지원 **datacenter** 역할이 **junior\_sysadmins**에 할당되도록 하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** [인벤토리 파일](#)을 생성했습니다.

- **Ansible** 인벤토리 파일은 `~/ <MyPlaybooks>/` 디렉터리에 있습니다.

#### 절차

1. `~/ <MyPlaybooks> /` 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. `/usr/share/doc/ansible-freeipa/playbooks/role/` 디렉터리에 있는 `role-member-group-present.yml` 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-group-present.yml
role-member-group-present-copy.yml
```

3. 편집할 `role-member-group-present-copy.yml` **Ansible** 플레이북 파일을 엽니다.
4. `iparole` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자의 암호로 설정합니다.
- 이름 변수를 할당하려는 역할의 이름으로 설정합니다.
- 그룹 변수를 그룹 이름으로 설정합니다.
- `action` 변수를 `member` 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no
```

```
tasks:
- iparole:
    ipaadmin_password: Secret123
    name: helpdesk
    group: junior_sysadmins
    action: member
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-group-present-copy.yml
```

#### 추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown](#) 파일을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/playbooks/iparole](#) 디렉터리에서 샘플 플레이북을 참조하십시오.

### 23.9. ANSIBLE을 사용하여 특정 사용자가 IDM RBAC 역할에 할당되지 않았는지 확인

**IdM(Identity Management)**에서 **RBAC(역할 기반 액세스 제어)**를 관리하는 시스템 관리자는 **RBAC** 역할이 회사 내의 다른 위치로 이동된 후 특정 사용자에게 할당되지 않도록 할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **user\_01** 및 **user\_02** 라는 사용자가 **helpRuntimeConfig** 역할에 할당되지 않도록 하는 방법을 설명합니다.

#### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 ~/ <MyPlaybooks>/ 디렉터리에 있습니다.

## 절차

1. ~/ <MyPlaybooks> / 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. /usr/share/doc/ansible-freeipa/playbooks/role/ 디렉터리에 있는 role-member-user-absent.yml 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-user-absent.yml role-member-user-absent-copy.yml
```

3. 편집할 role-member-user-absent-copy.yml Ansible 플레이북 파일을 엽니다.
4. iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin\_password 변수를 IdM 관리자의 암호로 설정합니다.
- 이름 변수를 할당하려는 역할의 이름으로 설정합니다.
- 사용자 목록을 사용자 이름으로 설정합니다.
- action 변수를 member 로 설정합니다.

- **state** 변수를 **absent** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
    - iparole:
      ipaadmin_password: Secret123
      name: helpdesk
      user
      - user_01
      - user_02
      action: member
      state: absent
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-user-absent-copy.yml
```

#### 추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/](#) 디렉토리의 **README-role Markdown** 파일을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/playbooks/iparole](#) 디렉터리에서 샘플 플레이북을 참조하십시오.

## 23.10. ANSIBLE을 사용하여 서비스가 IDM RBAC 역할의 멤버인지 확인

IdM(Identity Management)에서 RBAC(역할 기반 액세스 제어)를 관리하는 시스템 관리자는 IdM에 등록된 특정 서비스가 특정 역할의 멤버인지 확인할 수 있습니다. 다음 예제에서는 사용자 지정 `web_administrator` 역할이 `client01.idm.example.com` 서버에서 실행 중인 HTTP 서비스를 관리할 수 있도록 하는 방법을 설명합니다.

### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 `ansible-freeipa` 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 `~/ <MyPlaybooks>/` 디렉터리에 있습니다.
- IdM에 `web_administrator` 역할이 있습니다.
- IdM에 `HTTP/client01.idm.example.com@IDM.EXAMPLE.COM` 서비스가 있습니다.

### 절차

1. `~/ <MyPlaybooks> /` 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. `/usr/share/doc/ansible-freeipa/playbooks/role/` 디렉터리에 있는 `role-member-service-present.yml` 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-service-present-absent.yml role-member-service-present-copy.yml
```

3. 편집할 `role-member-service-present-copy.yml` Ansible 플레이북 파일을 엽니다.

4.

**iparole** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- 이름 변수를 할당하려는 역할의 이름으로 설정합니다.
- 서비스 목록을 서비스 이름으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: web_administrator
    service:
    - HTTP/client01.idm.example.com
    action: member
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-service-present-copy.yml
```

추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.

- [IdM의 역할을](#) 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-role Markdown** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리에서 샘플 플레이북을 참조하십시오.

### 23.11. ANSIBLE을 사용하여 호스트가 IDM RBAC 역할의 멤버인지 확인

**IdM(Identity Management)**에서 역할 기반 액세스 제어를 관리하는 시스템 관리자는 특정 호스트 또는 호스트 그룹이 특정 역할과 연결되도록 할 수 있습니다. 다음 예제에서는 사용자 지정 `web_administrator` 역할에서 HTTP 서비스가 실행 중인 `client01.idm.example.com` IdM 호스트를 관리할 수 있도록 하는 방법을 설명합니다.

#### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 [ansible-freeipa](#) 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 [Ansible 인벤토리 파일을](#) 생성했습니다.
- Ansible 인벤토리 파일은 `~/ <MyPlaybooks>/` 디렉터리에 있습니다.
- IdM에 `web_administrator` 역할이 있습니다.
- `client01.idm.example.com` 호스트는 IdM에 있습니다.

#### 절차

1. `~/ <MyPlaybooks> /` 디렉터리로 이동합니다.

■

```
$ cd ~/<MyPlaybooks>/
```

2.

/usr/share/doc/ansible-freeipa/playbooks/role/ 디렉터리에 있는 role-member-host-present.yml 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-host-present.yml role-member-host-present-copy.yml
```

3.

편집할 role-member-host-present-copy.yml Ansible 플레이북 파일을 엽니다.

4.

iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin\_password 변수를 IdM 관리자의 암호로 설정합니다.
- 이름 변수를 할당하려는 역할의 이름으로 설정합니다.
- 호스트 목록을 호스트 이름으로 설정합니다.

현재 예제에 대해 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: web_administrator
    host:
      - client01.idm.example.com
    action: member
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 Ansible 플레이북을 실행합니다.

-

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-host-present-copy.yml
```

#### 추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-role** Markdown 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리에서 샘플 플레이북을 참조하십시오.

### 23.12. ANSIBLE을 사용하여 호스트 그룹이 IDENTITY RBAC 역할의 멤버인지 확인

**IdM(Identity Management)**에서 역할 기반 액세스 제어를 관리하는 시스템 관리자는 특정 호스트 또는 호스트 그룹이 특정 역할과 연결되도록 할 수 있습니다. 다음 예제에서는 사용자 지정 `web_administrator` 역할이 HTTP 서비스가 실행 중인 IdM 호스트의 `web_servers` 그룹을 관리할 수 있도록 하는 방법을 설명합니다.

#### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 `ansible-freeipa` 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 [Ansible 인벤토리 파일](#)을 생성했습니다.
- Ansible 인벤토리 파일은 `~/<MyPlaybooks>/` 디렉터리에 있습니다.
- IdM에 `web_administrator` 역할이 있습니다.

- IdM에 **web\_servers** 호스트 그룹이 있습니다.

#### 절차

1. ~/ <MyPlaybooks> / 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. /usr/share/doc/ansible-freeipa/playbooks/role/ 디렉터리에 있는 **role-member-hostgroup-present.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-hostgroup-present.yml role-member-hostgroup-present-copy.yml
```

3. 편집할 **role-member-hostgroup-present-copy.yml** Ansible 플레이북 파일을 엽니다.
4. **iparole** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 IdM 관리자의 암호로 설정합니다.
- 이름 변수를 할당하려는 역할의 이름으로 설정합니다.
- **hostgroup** 목록을 호스트 그룹의 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: web_administrator
```

```
hostgroup:  
- web_servers  
action: member
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-hostgroup-present-copy.yml
```

#### 추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/](#) 디렉토리의 **README-role Markdown** 파일을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/playbooks/iparole](#) 디렉터리에서 샘플 플레이북을 참조하십시오.

## 24장. ANSIBLE 플레이북을 사용하여 RBAC 권한 관리

**RBAC**(역할 기반 액세스 제어)는 역할, 권한 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. 특히 대규모 회사에서 **RBAC**를 사용하면 개별 책임 영역이 있는 관리자의 계층적 시스템을 만드는 데 도움이 될 수 있습니다.

이 장에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에서 **RBAC** 권한을 관리하기 위한 다음 작업을 설명합니다.

- **Ansible을 사용하여 사용자 정의 RBAC 권한이 있는지 확인**
- **Ansible을 사용하여 사용자 정의 IdM RBAC 권한에 멤버 권한이 있는지 확인**
- **Ansible을 사용하여 IdM RBAC 권한에 권한이 포함되어 있지 않은지 확인**
- **Ansible을 사용하여 사용자 정의 IdM RBAC 권한 이름 변경**
- **Ansible을 사용하여 IdM RBAC 권한이 없는지 확인**

사전 요구 사항

- **RBAC의 개념과 원칙을 이해합니다.**

### 24.1. ANSIBLE을 사용하여 사용자 정의 IdM RBAC 권한이 있는지 확인

**IdM(Identity Management)** 역할 기반 액세스 제어(**RBAC**)에서 완전한 사용자 정의 권한을 얻으려면 단계를 진행해야 합니다.

1. 권한이 연결되어 있지 않은 권한을 생성합니다.
2. 선택한 권한을 권한에 추가합니다.

다음 절차에서는 나중에 권한을 추가할 수 있도록 **Ansible** 플레이북을 사용하여 빈 권한을 생성하는 방법을 설명합니다. 이 예제에서는 호스트 관리와 관련된 모든 **IdM** 권한을 결합하는 데 사용되는 **full\_host\_administration** 이라는 권한을 생성하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
- **Ansible** 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.

#### 절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/privilege/** 디렉터리에 있는 **privilege-present.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-present.yml privilege-present-copy.yml
```

3. 편집할 **privilege-present-copy.yml** **Ansible** 플레이북 파일을 엽니다.

4. **ipaprivilege** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin\_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 새 권한의 이름으로 설정합니다. **full\_host\_administration**.

●

필요한 경우 **description** 변수를 사용하여 권한을 설명합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Privilege present example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure privilege full_host_administration is present
      ipaprivilege:
        ipaadmin_password: Secret123
        name: full_host_administration
        description: This privilege combines all IdM permissions related to host
          administration
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory privilege-present-copy.yml
```

## 24.2. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한에 멤버 권한이 있는지 확인

**IdM(Identity Management)** 역할 기반 액세스 제어(**RBAC**)에서 완전한 사용자 정의 권한을 얻으려면 단계를 진행해야 합니다.

1.

권한이 연결되어 있지 않은 권한을 생성합니다.

2.

선택한 권한을 권한에 추가합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 이전 단계에서 생성된 권한에 권한을 추가하는 방법을 설명합니다. 이 예제에서는 호스트 관리와 관련된 모든 **IdM** 권한을 **full\_host\_administration** 이라는 권한에 추가하는 방법을 설명합니다. 기본적으로 권한은 호스트 등록, 호스트 관리자 및 호스트 그룹 관리자 권한 간에 배포됩니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.
- **full\_host\_administration** 권한이 있습니다. Ansible을 사용하여 권한을 생성하는 방법에 대한 자세한 내용은 **Using Ansible to ensure a custom IdM RBAC 권한이 있는지** 확인하십시오.

## 절차

1. ~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible-freeipa/playbooks/privilege/ 디렉터리에 있는 **privilege-member-present.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-member-present.yml  
privilege-member-present-copy.yml
```

3. 편집할 **privilege-member-present-copy.yml** Ansible 플레이북 파일을 엽니다.
4. **ipaprivilege** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin\_password** 변수를 IdM 관리자의 암호로 설정합니다.

- **name** 변수를 권한 이름으로 설정합니다.
- 권한 목록을 권한에 포함하려는 권한 이름으로 설정합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Privilege member present example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that permissions are present for the "full_host_administration"
      privilege
      ipaprivilege:
        ipaadmin_password: Secret123
        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. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory privilege-member-present-copy.yml
```

#### 24.3. ANSIBLE을 사용하여 IDM RBAC 권한에 권한이 포함되어 있지 않은지 확인

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 권한에서 권한을 제거하는 방법을 설명합니다. 이 예제에서는 기본 인증서 관리자 권한에서 **CA ACL** 권한을 무시하는 요청 인증서 제거 방법을 설명합니다. 예를 들어 관리자는 보안 위험을 고려합니다.

##### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.

##### 절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/privilege/** 디렉터리에 있는 **privilege-member-present.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-member-absent.yml
privilege-member-absent-copy.yml
```

3. 편집할 **privilege-member-absent-copy.yml** Ansible 플레이북 파일을 엽니다.

4. **ipaprivilege** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin\_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.
- 권한 에서 제거할 권한의 이름으로 권한 목록을 설정합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Privilege absent example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that the "Request Certificate ignoring CA ACLs" permission is absent
      from the "Certificate Administrators" privilege
      ipaprivilege:
        ipaadmin_password: Secret123
        name: Certificate Administrators
        permission:
          - "Request Certificate ignoring CA ACLs"
        action: member
        state: absent
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory privilege-member-absent-copy.yml
```

#### 24.4. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한 이름 변경

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 예를 들어 권한에서 몇 가지 권한을 제거했기 때문에 권한 이름 변경 방법을 설명합니다. 결과적으로 권한 이름이 더 이상 정확하지 않습니다. 이 예제에서 관리자는 **full\_host\_administration** 권한의 이름을 **limited\_host\_administration** 으로 변경합니다.

##### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.
- **full\_host\_administration** 권한이 있습니다. 권한을 추가하는 방법에 대한 자세한 내용은 **Using Ansible to ensure a custom IdM RBAC 권한이 있는지** 확인하십시오.

##### 절차

1.

~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2.

`/usr/share/doc/ansible-freeipa/playbooks/privilege/` 디렉터리에 있는 `privilege-present.yml` 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-present.yml rename-privilege.yml
```

3.

편집할 `rename-privilege.yml` Ansible 플레이북 파일을 엽니다.

4.

`ipaprivilege` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- `name` 변수를 현재 권한 이름으로 설정합니다.
- `rename` 변수를 추가하고 권한의 새 이름으로 설정합니다.
- `state` 변수를 추가하고 이름이 로 설정합니다.

5.

플레이북 자체의 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
---
- name: Rename a privilege
  hosts: ipaserver
  become: true
```

6.

플레이북의 작업 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
[...]
tasks:
- name: Ensure the full_host_administration privilege is renamed to
  limited_host_administration
  ipaprivilege:
  [...]
```

현재 예제에 대해 수정된 Ansible 플레이북 파일입니다.

■

```

---
- name: Rename a privilege
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure the full_host_administration privilege is renamed to
      limited_host_administration
      ipaprivilege:
        ipaadmin_password: Secret123
        name: full_host_administration
        rename: limited_host_administration
        state: renamed

```

7.

파일을 저장합니다.

8.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory rename-privilege.yml
```

## 24.5. ANSIBLE을 사용하여 IDM RBAC 권한이 없는지 확인

**IdM(Identity Management)**의 시스템 관리자는 **IdM** 역할 기반 액세스 제어를 사용자 지정할 수 있습니다. 다음 절차에서는 **Ansible** 플레이북을 사용하여 **RBAC** 권한이 없는지 확인하는 방법을 설명합니다. 이 예제에서는 **CA** 관리자 권한이 없는지 확인하는 방법을 설명합니다. 이 절차를 통해 관리자 관리자는 **IdM**에서 인증 기관을 관리할 수 있는 유일한 사용자가 됩니다.

### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
- **Ansible** 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.

### 절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/privilege/` 디렉터리에 있는 `privilege-absent.yml` 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-absent.yml privilege-absent-copy.yml
```

3. 편집할 `privilege-absent-copy.yml` Ansible 플레이북 파일을 엽니다.

4. `ipaprivilege` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 `IdM` 관리자의 암호로 설정합니다.
- 이름 변수를 제거하려는 권한의 이름으로 설정합니다.
- `state` 변수가 `absent` 로 설정되어 있는지 확인합니다.

5. 플레이북의 작업 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
[...]
tasks:
- name: Ensure privilege "CA administrator" is absent
  ipaprivilege:
  [...]
```

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Privilege absent example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure privilege "CA administrator" is absent
```

```
ipaprivilege:
  ipaadmin_password: Secret123
  name: CA administrator
  state: absent
```

6.

파일을 저장합니다.

7.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory privilege-absent-copy.yml
```

## 24.6. 추가 리소스

- [IdM의 권한 참조.](#)
- [IdM의 권한을 참조하십시오.](#)
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용 가능한 **README-privilege** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/ipaprivilege` 디렉터리에서 샘플 플레이북을 참조하십시오.

## 25장. ANSIBLE 플레이북을 사용하여 IDM에서 RBAC 권한 관리

**RBAC**(역할 기반 액세스 제어)는 역할, 권한 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. 특히 대규모 회사에서 **RBAC**를 사용하면 개별 책임 영역이 있는 관리자의 계층적 시스템을 만드는 데 도움이 될 수 있습니다.

이 장에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에서 **RBAC** 권한을 관리할 때 수행되는 다음 작업을 설명합니다.

- **Ansible을 사용하여 RBAC 권한이 있는지 확인**
- **Ansible을 사용하여 특성이 있는 RBAC 권한이 있는지 확인합니다.**
- **Ansible을 사용하여 RBAC 권한이 없는지 확인**
- **Ansible을 사용하여 속성이 IdM RBAC 권한의 멤버인지 확인**
- **Ansible을 사용하여 속성이 IdM RBAC 권한의 멤버가 아닌지 확인**
- **Ansible을 사용하여 IdM RBAC 권한 이름 변경**

사전 요구 사항

- **RBAC의 개념과 원칙을 이해합니다.**

### 25.1. ANSIBLE을 사용하여 RBAC 권한이 있는지 확인

**IdM(Identity Management)**의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 권한에 추가할 수 있도록 **Ansible** 플레이북을 사용하여 **IdM**에 권한이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한은 호스트에만 적용할 수 있습니다.
- 권한이 포함된 권한이 부여된 사용자에게는 해당 항목에 대해 가능한 다음 모든 작업을 수행할 수 있습니다.
  - 쓰기
  - 읽기
  - 검색
  - 비교
  - add
  - delete

#### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.

#### 절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2.

`/usr/share/doc/ansible-freeipa/playbooks/permission/` 디렉터리에 있는 `permission-present.yml` 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-present.yml
permission-present-copy.yml
```

3.

편집할 `permission-present-copy.yml` Ansible 플레이북 파일을 엽니다.

4.

`ipapermission` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- `name` 변수를 권한 이름으로 설정합니다.
- `object_type` 변수를 `host` 로 설정합니다.
- 올바른 변수를 모든 으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission present example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that the "MyPermission" permission is present
      ipapermission:
        ipaadmin_password: Secret123
        name: MyPermission
        object_type: host
        right: all
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory permission-present-copy.yml
```

**25.2. ANSIBLE**을 사용하여 특성이 있는 **RBAC** 권한이 있는지 확인합니다.

**IdM(Identity Management)**의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 권한에 추가할 수 있도록 **Ansible** 플레이북을 사용하여 **IdM**에 권한이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

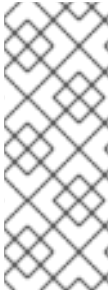
- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한은 호스트를 추가하는 데만 사용할 수 있습니다.
- 권한이 포함된 권한이 부여된 사용자에게는 호스트 항목에서 가능한 다음 모든 작업을 수행할 수 있습니다.
  - 쓰기
  - 읽기
  - 검색
  - 비교
  - add

○

**delete**

●

**MyPermission** 권한이 포함된 권한이 부여된 사용자가 생성한 호스트 항목은 설명 값을 가질 수 있습니다.



참고

권한을 생성할 때 지정할 수 있는 속성 유형은 **IdM LDAP** 스키마에 의해 제한되지 않습니다. 그러나 예를 들어 **object\_type** 이 호스트가 나중엔 **host** 인 경우 **attrs: car\_licence** 를 지정하면 권한을 수행하고 호스트에 특정 자동차 라이선스 값을 추가하려고 할 때 **ipa: ERROR: attribute "car-license"가 오류 메시지를 허용하지 않습니다.**

#### 사전 요구 사항

●

**IdM** 관리자 암호를 알고 있습니다.

●

**Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.

●

이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.

#### 절차

1.

**~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2.

**/usr/share/doc/ansible-freeipa/playbooks/permission/** 디렉터리에 있는 **permission-present.yml** 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-present.yml
permission-present-with-attribute.yml
```

3.

편집할 **permission-present-present-with-attribute.yml** **Ansible** 플레이북 파일을 엽니다.

4.

**ipapermission** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin\_password** 변수를 IdM 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.
- **object\_type** 변수를 **host** 로 설정합니다.
- 올바른 변수를 모든 으로 설정합니다.
- **attrs** 변수를 설명으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission present example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure that the "MyPermission" permission is present with an attribute
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      object_type: host
      right: all
      attrs: description
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory permission-present-with-attribute.yml
```

## 추가 리소스

- **RHEL 7의 *Linux* 도메인 ID, 인증 및 정책 가이드의 *사용자 및* 그룹 스키마**를 참조하십시오.

## 25.3. ANSIBLE을 사용하여 RBAC 권한이 없는지 확인

**IdM(Identity Management)**의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 권한에 추가할 수 없도록 **Ansible** 플레이북을 사용하여 **IdM**에 권한이 없는지 확인하는 방법을 설명합니다.

### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.

### 절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/permission/** 디렉터리에 있는 **permission-absent.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-absent.yml
permission-absent-copy.yml
```

3. 편집할 **permission-absent-copy.yml** **Ansible** 플레이북 파일을 엽니다.

4.

**ipapermission** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin\_password** 변수를 IdM 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission absent example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that the "MyPermission" permission is absent
      ipapermission:
        ipaadmin_password: Secret123
        name: MyPermission
        state: absent
```

5.

파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory permission-absent-copy.yml
```

## 25.4. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버인지 확인

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어(RBAC)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 속성이 IdM의 **RBAC** 권한 멤버인지 확인하는 방법을 설명합니다. 결과적으로 권한이 있는 사용자는 특성이 있는 항목을 생성할 수 있습니다.

이 예제에서는 **MyPermission** 권한이 포함된 사용자가 생성한 호스트 항목에 **gecos** 및 **description** 값을 가질 수 있는지 확인하는 방법을 설명합니다.



#### 참고

권한을 생성할 때 지정할 수 있는 속성 유형은 **IdM LDAP** 스키마에 의해 제한되지 않습니다. 그러나 예를 들어 **object\_type** 이 호스트가 나중에 **host** 인 경우 **attrs: car\_licence** 를 지정하면 권한을 수행하고 호스트에 특정 자동차 라이선스 값을 추가하려고 할 때 **ipa: ERROR: attribute "car-license"**가 오류 메시지를 허용하지 않습니다.

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.
- **MyPermission** 권한이 있습니다.

#### 절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/permission/** 디렉터리에 있는 **permission-member-present.yml** 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-member-present.yml permission-member-present-copy.yml
```

3. 편집할 **permission-member-present-copy.yml** Ansible 플레이북 파일을 엽니다.
4. **ipapermission** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- `ipaadmin_password` 변수를 **IdM** 관리자의 암호로 설정합니다.
- `name` 변수를 권한 이름으로 설정합니다.
- `attrs` 목록을 `description` 및 `gecos` 변수로 설정합니다.
- `action` 변수가 `member` 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission member present example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that the "gecos" and "description" attributes are present in
      "MyPermission"
      ipapermission:
        ipaadmin_password: Secret123
        name: MyPermission
        attrs:
          - description
          - geccos
        action: member
```

5. 파일을 저장합니다.
6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory permission-member-present-copy.yml
```

## 25.5. ANSIBLE을 사용하여 속성이 **IDM RBAC** 권한의 멤버가 아닌지 확인

**IdM(Identity Management)**의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 속성이 **IdM**의 **RBAC** 권한 멤버가 아닌지 확인하는 방법을 설명합니다. 결과적으로 권한이 있는 사용자가 **IdM LDAP**에 항목을 생성하면 해당 항목에 속성과 연결된 값을 가질 수 없습니다.

이 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한이 포함된 권한이 있는 사용자가 생성한 호스트 항목은 **description** 속성을 가질 수 없습니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉토리를 **생성하고 구성** 했다고 가정합니다.
- **MyPermission** 권한이 있습니다.

절차

1. **~/MyPlaybooks/** 디렉토리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/permission/** 디렉토리에 있는 **permission-member-absent.yml** 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-member-absent.yml permission-member-absent-copy.yml
```

3. 편집할 **permission-member-absent-copy.yml Ansible** 플레이북 파일을 엽니다.

4. **ipapermission** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin\_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.
- **attrs** 변수를 설명으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.
- **state** 변수가 **absent**로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission absent example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that an attribute is not a member of "MyPermission"
      ipapermission:
        ipaadmin_password: Secret123
        name: MyPermission
        attrs: description
        action: member
        state: absent
```

5. 파일을 저장합니다.

6.

플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory permission-member-absent-copy.yml
```

## 25.6. ANSIBLE을 사용하여 IDM RBAC 권한 이름 변경

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 권한의 이름을 바꾸는 방법을 설명합니다. 이 예제에서는 **MyPermission**의 이름을 **MyNewPermission**으로 변경하는 방법을 설명합니다.

### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.
- IdM에 **MyPermission**이 있습니다.
- IdM에는 **MyNewPermission**이 없습니다.

### 절차

1.

**~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2.

**/usr/share/doc/ansible-freeipa/playbooks/permission/** 디렉터리에 있는 **permission-renamed.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-renamed.yml
permission-renamed-copy.yml
```

3. 편집할 **permission-renamed-copy.yml** Ansible 플레이북 파일을 엽니다.

4. **ipapermission** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin\_password** 변수를 IdM 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission present example
  hosts: ipaserver
  become: true

  tasks:
  - name: Rename the "MyPermission" permission
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      rename: MyNewPermission
      state: renamed
```

5. 파일을 저장합니다.

6. 플레이북 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory permission-renamed-copy.yml
```

## 25.7. 추가 리소스

- [IdM의 권한](#)을 참조하십시오.
- [IdM의 권한](#) 참조.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용 가능한 **README-permission** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/ipapermission` 디렉터리에서 샘플 플레이북을 참조하십시오.

## 26장. ID 보기를 사용하여 IDM 클라이언트의 사용자 속성 값 덮어쓰기

**IdM(Identity Management)** 사용자가 **IdM LDAP** 서버에 저장된 사용자 또는 그룹 속성 중 일부를 재정의하려는 경우(예: 인증에 사용되는 로그인 이름, 홈 디렉터리, 인증서 또는 **SSH** 키)를 재정의하면 **IdM** 관리자가 **IdM ID** 뷰를 사용하여 특정 **IdM** 클라이언트의 이러한 값을 재정의할 수 있습니다. 예를 들어 사용자가 **IdM**에 로그인하는 데 가장 일반적으로 사용하는 **IdM** 클라이언트의 사용자에게 대해 다른 홈 디렉터리를 지정할 수 있습니다.

이 장에서는 **IdM**에 **IdM**에 등록된 호스트의 **IdM** 사용자와 관련된 **POSIX** 속성 값을 클라이언트로 재정의하는 방법을 설명합니다. 특히 이 장에서는 사용자 로그인 이름과 홈 디렉토리를 재정의하는 방법을 설명합니다.

이 장에는 다음 섹션이 포함되어 있습니다.

- [ID 보기](#)
- [SSSD 성능에 대한 ID 보기의 부정적인 영향](#)
- [ID 보기의 속성을 재정의할 수 있습니다.](#)
- [ID 보기 명령에 대한 도움말 가져오기](#)
- [ID 보기를 사용하여 특정 호스트에서 IdM 사용자의 로그인 이름 덮어쓰기](#)
- [IdM ID 보기 수정](#)
- [IdM 클라이언트의 IdM 사용자 홈 디렉토리를 덮어쓰는 ID 보기 추가](#)
- [IdM 호스트 그룹에 ID 보기 적용](#)

### 26.1. ID 보기

IdM(Identity Management)의 ID 보기는 다음 정보를 지정하는 IdM 클라이언트 쪽 보기입니다.

- 중앙에서 정의한 **POSIX** 사용자 또는 그룹 특성에 대한 새로운 값
- 새 값이 적용되는 클라이언트 호스트 또는 호스트입니다.

ID 뷰에는 하나 이상의 덮어쓰기가 포함되어 있습니다. 재정의는 중앙에서 정의한 **POSIX** 특성 값을 구체적으로 대체하는 것입니다.

IdM 서버에서 중앙에서 IdM 클라이언트의 ID 보기만 정의할 수 있습니다. IdM 클라이언트에 대한 클라이언트 측 재정의는 로컬에서 구성할 수 없습니다.

예를 들어 ID 보기를 사용하여 다음 목표를 달성할 수 있습니다.

- 다양한 환경에 대해 서로 다른 특성 값을 정의합니다. 예를 들어 IdM 관리자 또는 다른 IdM 사용자가 서로 다른 IdM 클라이언트에 다른 홈 디렉토리를 갖도록 허용할 수 있습니다. `/home/encrypted/username` 을 하나의 IdM 클라이언트 및 다른 클라이언트의 `/dropbox/username` 에서 이 사용자의 홈 디렉터리로 구성할 수 있습니다. 이 상황에서 ID 뷰를 사용하는 것은 또는 `alternative_homedir` , `override_homedir` 또는 클라이언트의 `/etc/sss/sss.conf` 파일의 기타 홈 디렉터리 변수를 변경하는 것이 모든 사용자에게 영향을 미칩니다. 예제 프로시저 는 [IdM 클라이언트의 IdM 사용자 홈 디렉터를 덮어쓰려면 ID 보기 추가](#) 를 참조하십시오.
- 이전에 생성된 속성 값을 사용자의 UID 재정의와 같은 다른 값으로 교체합니다. 이 기능은 LDAP 측에서 수행하기 어려운 시스템 전체 변경을 달성하려는 경우 유용할 수 있습니다(예: IdM 사용자의 UID 1009 만들기). IdM 사용자 UID를 생성하는 데 사용되는 IdM ID 범위는 1000 또는 10000으로 시작하지 않습니다. IdM 사용자가 모든 IdM 클라이언트에서 UID 1009인 로컬 사용자를 가장하는 이유가 있는 경우 ID 보기를 사용하여 IdM에서 사용자가 생성될 때 생성된 이 IdM 사용자의 UID를 재정의할 수 있습니다.



중요

IdM 서버는 아닌 IdM 클라이언트에만 ID 뷰를 적용할 수 있습니다.

추가 리소스

- **Active Directory** 사용자를 위한 **ID** 보기 사용
- **SSSD** 클라이언트 측 뷰

## 26.2. SSSD 성능에 대한 ID 보기의 부정적인 영향

ID 보기를 정의할 때 **IdM**은 **IdM** 서버의 **SSSD(System Security Services Daemon)** 캐시에 필요한 덮어쓰기 값을 배치합니다. 그런 다음 **IdM** 클라이언트에서 실행 중인 **SSSD**는 서버 캐시에서 재정의 값을 검색합니다.

특정 최적화 및 ID 뷰를 동시에 실행할 수 없기 때문에 ID 보기를 적용하면 **SSSD(System Security Services Daemon)** 성능에 부정적인 영향을 미칠 수 있습니다. 예를 들어 ID 보기에서 **SSSD**가 서버에서 그룹을 조회하는 프로세스를 최적화하지 못하도록 합니다.

- ID 뷰의 경우 **SSSD**는 그룹 이름을 재정의하는 경우 반환된 그룹 멤버 이름 목록의 모든 멤버를 확인해야 합니다.
- ID 보기를 사용하지 않으면 **SSSD**에서 그룹 개체의 **member** 속성에서만 사용자 이름을 수집할 수 있습니다.

**SSSD** 캐시가 비어 있거나 캐시를 지워지면 이 부정적인 영향은 모든 항목을 무효화합니다.

## 26.3. ID 보기의 속성을 재정의할 수 있습니다.

ID 보기는 사용자 및 그룹 ID 재정의로 구성됩니다. 덮어쓰기는 새 **POSIX** 특성 값을 정의합니다.

사용자 및 그룹 ID 덮어쓰기는 다음 **POSIX** 특성에 대한 새 값을 정의할 수 있습니다.

### 사용자 속성

- 로그인 이름(**uid**)
- **GECOS** 항목(**Gecos**)

- **UID 번호(uidNumber)**
- **GID 번호(gidNumber)**
- **로그인 셸(loginShell)**
- **홈 디렉토리(homeDirectory)**
- **SSH 공개 키(ipaSshPubkey)**
- **인증서(사용자 인증서)**

#### 그룹 속성

- **그룹 이름(cncn)**
- **그룹 GID 번호(gidNumber)**

### 26.4. ID 보기 명령에 대한 도움말 가져오기

**IdM CLI**(명령줄 인터페이스)의 **IdM(Identity Management)** ID 보기와 관련된 명령에 대한 도움말을 얻을 수 있습니다.

#### 사전 요구 사항

- **IdM** 사용자를 위한 **Kerberos** 티켓이 있습니다.

#### 절차

- **ID** 보기 및 덮어쓰기를 관리하는 데 사용되는 모든 명령을 표시하려면 다음을 수행합니다.

```
$ 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

[...]

- 

특정 명령에 대한 자세한 도움말을 표시하려면 명령에 **--help** 옵션을 추가합니다.

```
$ 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

[...]

## 26.5. ID 보기를 사용하여 특정 호스트에서 IDM 사용자의 로그인 이름 덮어쓰기

이 섹션에서는 **IdM(Identity Management)** 시스템 관리자가 특정 **IdM** 사용자와 관련된 **POSIX** 특성 값을 재정의하는 특정 **IdM** 클라이언트에 대한 **ID** 보기를 생성하는 방법을 설명합니다. 이 절차에서는 **ID** 보기의 예제를 사용하여 **idm\_user** 라는 **IdM** 사용자가 **user\_1234** 로그인 이름을 사용하여 **host1** 이라는 **IdM** 클라이언트에 로그인할 수 있도록 합니다.

### 사전 요구 사항

- 

**IdM** 관리자로 로그인되어 있습니다.

### 절차

- 1.

새 **ID** 보기 만들기 예를 들어 **example\_for\_host1** 이라는 **ID** 뷰를 생성하려면 다음을 수행합니다.

```
$ ipa idview-add example_for_host1
```

```
-----
Added ID View "example_for_host1"
-----
```

```
ID View Name: example_for_host1
```

- 2.

**example\_for\_host1** **ID** 보기에 사용자 재정의 추가합니다. 사용자 로그인을 덮어쓰려면 다

음을 수행합니다.

- **ipa idoverrideuser-add** 명령을 입력합니다.
- ID 보기의 이름 추가
- 앵커라고도 하는 사용자 이름 추가
- **--login** 옵션을 추가합니다.

```
$ 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
```

사용 가능한 옵션 목록은 **ipa idoverrideuser-add --help**를 실행합니다.



참고

**ipa idoverrideuser-add --certificate** 명령은 지정된 ID 보기의 계정의 기존 인증서를 모두 대체합니다. 추가 인증서를 추가하려면 대신 **ipa idoverrideuser-add-cert** 명령을 사용하십시오.

```
$ ipa idoverrideuser-add-cert example_for_host1 user --
certificate="MIIEATCC..."
```

3. 선택 사항: **ipa idoverrideuser-mod** 명령을 사용하여 기존 사용자 재정의에 대한 새 속성 값을 지정할 수 있습니다.
4. **example\_for\_host1** 을 **host1.idm.example.com** 호스트에 적용합니다.

```
$ 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



#### 참고

**ipa idview-apply** 명령은 **--hostgroups** 옵션도 허용합니다. 옵션은 지정된 호스트 그룹에 속하는 호스트에 ID 뷰를 적용하지만 ID 뷰를 호스트 그룹 자체와 연결하지 않습니다. 대신 **--hostgroups** 옵션은 지정된 호스트 그룹의 멤버를 확장하고 **--hosts** 옵션을 개별적으로 모든 그룹에 적용합니다.

즉, 나중에 호스트를 호스트 그룹에 추가하면 ID 보기가 새 호스트에 적용되지 않습니다.

5. 새 구성을 **host1.idm.example.com** 시스템에 즉시 적용하려면 다음을 수행하십시오.

- a. SSH를 사용하여 시스템에 **root**로 연결합니다.

```
$ ssh root@host1
Password:
```

- b. SSSD 캐시를 지웁니다.

```
root@host1 ~]# sss_cache -E
```

- c. SSSD 데몬을 다시 시작합니다.

```
root@host1 ~]# systemctl restart sssd
```

#### 검증 단계

- **user\_1234**의 인증 정보가 있는 경우 이를 사용하여 **host1**에서 IdM에 로그인할 수 있습니다.

1. 로그인 이름으로 **user\_1234**를 사용하여 **host1**에 SSH로 연결합니다.

```
[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.

작업 디렉터리를 표시합니다.

```
[user_1234@host1 ~]$ pwd
/home/idm_user/
```

•

또는 **host1** 에 **root** 인증 정보가 있는 경우 이를 사용하여 **id** 명령의 출력을 **id\_user** 및 **user\_1234** 에 확인할 수 있습니다.

```
[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)
```

## 26.6. IDM ID 보기 수정

**IdM(Identity Management)**의 **ID** 보기는 특정 **IdM** 사용자와 관련된 **POSIX** 특성 값을 재정의합니다. 이 섹션에서는 기존 **ID** 보기를 수정하는 방법에 대해 설명합니다. 특히, **idm\_user** 사용자가 **host1.idm\_user/**가 **IdM** 클라이언트에서 **/home/idm\_user/** 대신 **/home/user\_1234/** 디렉터리를 사용자 홈 디렉터리로 사용하도록 **ID** 보기를 수정하는 방법을 설명합니다.

### 사전 요구 사항

•

**host1.idm.example.com** 에 대한 루트 액세스 권한이 있습니다.

•

필요한 권한이 있는 사용자로 로그인했습니다(예: **admin** ).

•

**host1 IdM** 클라이언트에 적용되는 **idm\_user** 에 대해 구성된 **ID** 보기가 있습니다.

### 절차

1.

**root**로 **host1.idm.example.com** 에서 사용자 홈 디렉터리로 **idm\_user** 를 사용할 디렉터리를 만듭니다.

```
[root@host1 /]# mkdir /home/user_1234/
```

2.

디렉터리의 소유권을 변경합니다.

```
[root@host1 /]# chown idm_user:idm_user /home/user_1234/
```

3.

ID 뷰가 현재 적용된 호스트를 포함하여 ID 보기를 표시합니다. 이름이 **example\_for\_host1** 인 ID 뷰를 표시하려면 다음을 수행합니다.

```
$ 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
```

출력은 현재 ID 보기가 **host1.idm.example.com** 에 적용됨을 보여줍니다.

4.

**example\_for\_host1** ID 보기의 사용자 재정의의 수정합니다. 사용자 홈 디렉터리를 재정의하려면 다음을 수행합니다.

- **ipa idoverrideuser-add** 명령을 입력합니다.

- ID 보기의 이름 추가

- 앵커라고도 하는 사용자 이름 추가

- **--homedir** 옵션을 추가합니다.

```
$ 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/
```

사용 가능한 옵션 목록은 **ipa idoverrideuser-mod --help** 를 실행합니다.

5. 새 구성을 **host1.idm.example.com** 시스템에 즉시 적용하려면 다음을 수행하십시오.

- a. **SSH**를 사용하여 시스템에 **root**로 연결합니다.

```
$ ssh root@host1
Password:
```

- b. **SSSD** 캐시를 지웁니다.

```
root@host1 ~]# sss_cache -E
```

- c. **SSSD** 데몬을 다시 시작합니다.

```
root@host1 ~]# systemctl restart sssd
```

#### 검증 단계

1. **idm\_user** 로 **host1** 에 **SSH** 연결을 수행합니다.

```
[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. 작업 디렉터리를 출력합니다.

```
[user_1234@host1 ~]$ pwd
/home/user_1234/
```

#### 추가 리소스

- [기본 신뢰 뷰를 수정하여 AD 사용자의 글로벌 속성 정의](#)

### 26.7. IDM 클라이언트의 IDM 사용자 홈 디렉터리를 덮어쓰는 ID 보기 추가

**IdM(Identity Management)**의 ID 보기는 특정 **IdM** 사용자와 관련된 **POSIX** 특성 값을 재정의합니다. 이 섹션에서는 이름이 **host1** 인 **IdM** 클라이언트의 **idm\_user** 에 적용되는 ID 보기를 생성하여 사용자가

`/home/idm_user/` 가 아닌 사용자 홈 디렉터리로 `/home/user_1234/` 디렉터를 사용할 수 있도록 하는 방법을 설명합니다.

#### 사전 요구 사항

- `host1.idm.example.com` 에 대한 루트 액세스 권한이 있습니다.
- 필요한 권한이 있는 사용자로 로그인했습니다(예: `admin` ).

#### 절차

1. `root`로 `host1.idm.example.com` 에서 사용자 홈 디렉터리로 `idm_user` 를 사용할 디렉터를 만듭니다.

```
[root@host1 /]# mkdir /home/user_1234/
```

2. 디렉터리의 소유권을 변경합니다.

```
[root@host1 /]# chown idm_user:idm_user /home/user_1234/
```

3. ID 보기를 생성합니다. 예를 들어 `example_for_host1` 이라는 ID 뷰를 생성하려면 다음을 수행합니다.

```
$ ipa idview-add example_for_host1
-----
Added ID View "example_for_host1"
-----
ID View Name: example_for_host1
```

4. `example_for_host1` ID 보기에 사용자 재정의의를 추가합니다. 사용자 홈 디렉터를 재정의하려면 다음을 수행합니다.

- `ipa idoverrideuser-add` 명령을 입력합니다.
- ID 보기의 이름 추가

- 앵커라고도 하는 사용자 이름 추가
- `--homedir` 옵션을 추가합니다.

```
$ 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. `example_for_host1` 을 `host1.idm.example.com` 호스트에 적용합니다.

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

#### 참고

`ipa idview-apply` 명령은 `--hostgroups` 옵션도 허용합니다. 옵션은 지정된 호스트 그룹에 속하는 호스트에 ID 뷰를 적용하지만 ID 뷰를 호스트 그룹 자체와 연결하지 않습니다. 대신 `--hostgroups` 옵션은 지정된 호스트 그룹의 멤버를 확장하고 `--hosts` 옵션을 개별적으로 모든 그룹에 적용합니다.

즉, 나중에 호스트를 호스트 그룹에 추가하면 ID 보기가 새 호스트에 적용되지 않습니다.

6. 새 구성을 `host1.idm.example.com` 시스템에 즉시 적용하려면 다음을 수행하십시오.

- a. SSH를 사용하여 시스템에 `root`로 연결합니다.

```
$ ssh root@host1
Password:
```

b.

SSSD 캐시를 지웁니다.

```
root@host1 ~]# sss_cache -E
```

c.

SSSD 데몬을 다시 시작합니다.

```
root@host1 ~]# systemctl restart sssd
```

#### 검증 단계

1.

idm\_user 로 host1 에 SSH 연결을 수행합니다.

```
[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.

작업 디렉토리를 출력합니다.

```
[idm_user@host1 /]$ pwd
/home/user_1234/
```

#### 추가 리소스

•

[ID 보기를 사용하여 IdM 클라이언트의 AD 사용자의 기본 신뢰 보기 속성 덮어쓰기](#)

## 26.8. IDM 호스트 그룹에 ID 보기 적용

`ipa idview-apply` 명령은 `--hostgroups` 옵션을 허용합니다. 그러나 옵션은 현재 지정된 호스트 그룹에 속하는 호스트에 ID 보기를 적용하는 일회성 작업으로 작동하지만 ID 뷰를 호스트 그룹 자체와 동적으로 연관하지는 않습니다. `--hostgroups` 옵션은 지정된 호스트 그룹의 멤버를 확장하고 `--hosts` 옵션을 개별적으로 해당 그룹에 적용합니다.

나중에 호스트 그룹에 새 호스트를 추가하는 경우 `--hosts` 옵션과 함께 `ipa idview-apply` 명령을 사용하여 ID 보기를 새 호스트에 수동으로 적용해야 합니다.

마찬가지로 호스트 그룹에서 호스트를 제거해도 제거 후에도 ID 보기가 호스트에 계속 할당됩니다. 제거된 호스트에서 ID 보기를 적용 취소하려면 `ipa idview-unapply id_view_name --hosts=name_of_the_removed_host` 명령을 실행해야 합니다.

이 섹션에서는 다음 목표를 달성하는 방법에 대해 설명합니다.

1. 호스트 그룹을 생성하고 여기에 호스트를 추가하는 방법.
2. 호스트 그룹에 ID 보기를 적용하는 방법.
3. 호스트 그룹에 새 호스트를 추가하고 ID 보기를 새 호스트에 적용하는 방법.

#### 사전 요구 사항

- 호스트 그룹에 적용할 ID 보기가 IdM에 있는지 확인합니다. 예를 들어 AD 사용자의 GID를 재정의하는 ID 보기를 생성하려면 ID 뷰를 사용하여 IdM 클라이언트의 AD 사용자에 대한 기본 신뢰 보기 속성 덮어쓰기를 참조하십시오.

#### 절차

1. 호스트 그룹을 생성하고 여기에 호스트를 추가합니다.
  - a. 호스트 그룹을 생성합니다. 예를 들어 이름이 **baltimore** 인 호스트 그룹을 생성하려면 다음을 수행합니다.

```
[root@server ~]# ipa hostgroup-add --desc="Baltimore hosts" baltimore
-----
Added hostgroup "baltimore"
-----
Host-group: baltimore
Description: Baltimore hosts
```

- b. 호스트 그룹에 호스트를 추가합니다. 예를 들어 **host102** 및 **host103** 을 **baltimore** 호스트 그룹에 추가하려면 다음을 수행합니다.

```
[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.

호스트 그룹의 호스트에 ID 보기를 적용합니다. 예를 들어 **example\_for\_host1** ID 보기를 **baltimore** 호스트 그룹에 적용하려면 다음을 수행합니다.

```
[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.

새 호스트를 호스트 그룹에 추가하고 ID 보기를 새 호스트에 적용합니다.

a.

호스트 그룹에 새 호스트를 추가합니다. 예를 들어 **somehost.idm.example.com** 호스트를 **baltimore** 호스트 그룹에 추가하려면 다음을 수행합니다.

```
[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.

선택적으로 ID 보기 정보를 표시합니다. 예를 들어 **example\_for\_host1** ID 보기에 대한 세부 정보를 표시하려면 다음을 수행합니다.

```
[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
```

출력에서 ID 보기가 **baltimore** 호스트 그룹의 새로 추가된 호스트인 **somehost.idm.example.com** 에 적용되지 않음을 보여줍니다.

C.

새 호스트에 ID 뷰를 적용합니다. 예를 들어 **example\_for\_host1** ID 보기를 **somehost.idm.example.com**에 적용하려면 다음을 수행합니다.

```
[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
-----
```

검증 단계

•

ID 보기 정보를 다시 표시합니다.

```
[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
```

출력에 ID 보기가 **baltimore** 호스트 그룹에 새로 추가된 호스트인 **somehost.idm.example.com**에 적용됨을 보여줍니다.

## 26.9. NIS 도메인을 IDENTITY MANAGEMENT로 마이그레이션

ID 뷰를 사용하여 기존 호스트에 대한 호스트 특정 **UID** 및 **GID**를 설정하여 **NIS** 도메인을 **IdM**으로 마이그레이션할 때 파일 및 디렉터리의 권한 변경을 방지할 수 있습니다.

사전 요구 사항

•

**kinit admin** 명령을 사용하여 자신을 관리자으로 인증했습니다.

절차

1.

**IdM** 도메인에 사용자 및 그룹을 추가합니다.

a.

**ipa user-add** 명령을 사용하여 사용자를 생성합니다. 자세한 내용은 다음을 참조하십시오. [IdM에 사용자 추가](#)를 참조하십시오.

b.

**ipa group-add** 명령을 사용하여 그룹을 생성합니다. 자세한 내용은 [IdM에 그룹 추가](#)를 참조하십시오.

2.

사용자 생성 중에 생성된 ID를 덮어씁니다.

a.

**ipa idview-add** 명령을 사용하여 새 ID 보기를 생성합니다. 자세한 내용은 [ID 보기 명령에 대한 도움말](#)을 참조하십시오.

b.

각각 **ipa idoverrideuser-add** 및 **idoverridegroup-add**를 사용하여 ID 보기에 사용자 및 그룹의 ID 재정의의 추가합니다.

3.

**ipa idview-apply** 명령을 사용하여 ID 보기를 특정 호스트에 할당합니다.

4.

**NIS 도메인 해제.**

## 검증

1.

모든 사용자와 그룹이 ID 보기에 올바르게 추가되었는지 확인하려면 **ipa idview-show** 명령을 사용합니다.

```
$ ipa idview-show example-view
ID View Name: example-view
User object overrides: example-user1
Group object overrides: example-group
```

## 27장. ACTIVE DIRECTORY 사용자를 위한 ID 보기 사용

ID 보기를 사용하여 **IdM-AD** 신뢰 환경에서 **AD(Active Directory)** 사용자의 **POSIX** 속성에 대한 새 값을 지정할 수 있습니다.

기본적으로 **IdM**은 기본 신뢰 보기를 모든 **AD** 사용자에게 적용합니다. 개별 **IdM** 클라이언트에 추가 ID 뷰를 구성하여 수신하는 **POSIX** 속성별 사용자를 추가로 조정할 수 있습니다.

### 27.1. 기본 신뢰 보기의 작동 방식

**Default Trust View** 는 항상 신뢰 기반 설정의 **AD** 사용자 및 그룹에 적용되는 기본 ID 보기입니다. **ipa-adtrust-install** 명령을 사용하여 신뢰를 설정할 때 자동으로 생성되며 삭제할 수 없습니다.



참고

**Default Trust View**는 **IdM** 사용자 및 그룹이 아닌 **AD** 사용자 및 그룹에 대한 덮어쓰기만 허용합니다.

기본 신뢰 보기를 사용하여 **AD** 사용자 및 그룹에 대해 사용자 지정 **POSIX** 특성을 정의하여 **AD**에 정의된 값을 재정의할 수 있습니다.

표 27.1. 기본 신뢰 보기 적용

|       | AD의 값   | 기본 신뢰 보기 | 결과      |
|-------|---------|----------|---------|
| login | ad_user | ad_user  | ad_user |
| UID   | 111     | 222      | 222     |
| GID   | 111     | (값 없음)   | 111     |

**IdM** 클라이언트의 기본 신뢰 보기를 재정의하도록 추가 ID 뷰를 구성할 수도 있습니다. **IdM**은 기본 신뢰 보기 상단에 호스트별 ID 보기의 값을 적용합니다.

•

속성이 호스트별 ID 보기에 정의된 경우 **IdM**은 이 ID 보기의 값을 적용합니다.

- 속성이 호스트별 ID 보기에 정의되지 않은 경우 IdM은 기본 신뢰 보기의 값을 적용합니다.

표 27.2. 기본 신뢰 보기 상단에 호스트별 ID 보기 적용

|       | AD의 값   | 기본 신뢰 보기 | 호스트별 ID 보기 | 결과      |
|-------|---------|----------|------------|---------|
| login | ad_user | ad_user  | (값 없음)     | ad_user |
| UID   | 111     | 222      | 333        | 333     |
| GID   | 111     | (값 없음)   | 333        | 333     |



## 참고

호스트별 ID 보기만 적용하여 IdM 클라이언트의 기본 신뢰 보기를 덮어쓸 수 있습니다. IdM 서버와 복제본은 항상 **Default Trust View**의 값을 적용합니다.

## 추가 리소스

- [ID 보기를 사용하여 IdM 클라이언트의 사용자 속성 값 덮어쓰기](#)

## 27.2. 기본 신뢰 뷰를 수정하여 AD 사용자의 글로벌 속성 정의

전체 IdM 배포 전반에 걸쳐 **AD(Active Directory)** 사용자에게 대한 **POSIX** 속성을 재정의하려면 **Default Trust View**에서 해당 사용자의 항목을 수정하십시오. 이 절차에서는 **AD** 사용자 **ad\_user@ad.example.com**의 **GID**를 **732000006**으로 설정합니다.

## 사전 요구 사항

- 신뢰할 수 있는 컨트롤러 또는 신뢰 에이전트인 IdM 서버와 작업하고 있습니다.
- IdM 관리자가 인증되었습니다.

## 절차

1. IdM 관리자로서 **GID** 번호를 **732000006**으로 변경하는 기본 신뢰 보기에서 **AD** 사용자에게 대한 **ID**를 생성합니다.

```
# ipa idoverrideuser-add 'Default Trust View' ad_user@ad.example.com --
gidnumber=732000006
```

2.

모든 IdM 서버 및 클라이언트의 SSSD 캐시에서 **ad\_user@ad.example.com** 사용자의 항목을 지웁니다. 이렇게 하면 오래된 데이터가 제거되고 새로운 덮어쓰기 값을 적용할 수 있습니다.

```
# sssctl cache-expire -u ad_user@ad.example.com
```

검증

- **ad\_user@ad.example.com** 사용자의 정보를 검색하여 GID가 업데이트된 값을 반영하는지 확인합니다.

```
# 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)
```

### 27.3. ID 보기를 사용하여 IDM 클라이언트의 AD 사용자의 기본 신뢰 보기 속성 덮어쓰기

AD(Active Directory) 사용자에 대한 기본 신뢰 보기에서 일부 POSIX 속성을 재정의할 수 있습니다. 예를 들어 특정 IdM 클라이언트에 AD 사용자에게 다른 GID를 제공해야 할 수 있습니다. ID 보기를 사용하여 AD 사용자에 대한 기본 신뢰 보기의 값을 재정의하여 단일 호스트에 적용할 수 있습니다. 이 절차에서는 **host1.idm.example.com** IdM 클라이언트에 있는 **ad\_user@ad.example.com** AD 사용자의 GID를 **732001337**으로 설정하는 방법을 설명합니다.

사전 요구 사항

- **host1.idm.example.com** IdM 클라이언트에 대한 루트 액세스 권한이 있습니다.
- 필요한 권한이 있는 사용자로 로그인했습니다(예: **admin** 사용자).

절차

1.

ID 보기를 생성합니다. 예를 들어 **example\_for\_host1**이라는 ID 뷰를 생성하려면 다음을 수행합니다.

```
$ ipa idview-add example_for_host1
-----
Added ID View "example_for_host1"
-----
ID View Name: example_for_host1
```

2.

**example\_for\_host1** ID 보기에 사용자 재정의의 추가합니다. 사용자의 **GID**를 덮어쓰려면 다음을 수행합니다.

- **ipa idoverrideuser-add** 명령을 입력합니다.
- ID 보기의 이름 추가
- 앵커라고도 하는 사용자 이름 추가
- **--gidnumber=** 옵션을 추가합니다.

```
$ 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.

**example\_for\_host1** 을 **host1.idm.example.com** IdM 클라이언트에 적용합니다.

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

#### 참고

**ipa idview-apply** 명령은 **--hostgroups** 옵션도 허용합니다. 옵션은 지정된 호스트 그룹에 속하는 호스트에 ID 뷰를 적용하지만 ID 뷰를 호스트 그룹 자체와 연결하지 않습니다. 대신 **--hostgroups** 옵션은 지정된 호스트 그룹의 멤버를 확장하고 **--hosts** 옵션을 개별적으로 모든 그룹에 적용합니다.

즉, 나중에 호스트를 호스트 그룹에 추가하면 ID 보기가 새 호스트에 적용되지 않습니다.

4.

**host1.idm.example.com** IdM 클라이언트의 SSSD 캐시에서 **ad\_user@ad.example.com** 사용자의 항목을 지웁니다. 이렇게 하면 오래된 데이터가 제거되고 새로운 덮어쓰기 값을 적용할 수 있습니다.

```
[root@host1 ~]# sssctl cache-expire -u ad_user@ad.example.com
```

#### 검증 단계

1.

**ad\_user@ad.example.com** 로 **host1** 에 SSH 연결을 수행합니다.

```
[root@r8server ~]# ssh ad_user@ad.example.com@host1.idm.example.com
```

2.

**ad\_user@ad.example.com** 사용자의 정보를 검색하여 **GID**가 업데이트된 값을 반영하는지 확인합니다.

```
[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)
```

### 27.4. IDM 호스트 그룹에 ID 보기 적용

**ipa idview-apply** 명령은 **--hostgroups** 옵션을 허용합니다. 그러나 옵션은 현재 지정된 호스트 그룹에 속하는 호스트에 ID 보기를 적용하는 일회성 작업으로 작동하지만 ID 뷰를 호스트 그룹 자체와 동적으로 연관하지는 않습니다. **--hostgroups** 옵션은 지정된 호스트 그룹의 멤버를 확장하고 **--hosts** 옵션을 개별적으로 해당 그룹에 적용합니다.

나중에 호스트 그룹에 새 호스트를 추가하는 경우 **--hosts** 옵션과 함께 **ipa idview-apply** 명령을 사용하여 ID 보기를 새 호스트에 수동으로 적용해야 합니다.

마찬가지로 호스트 그룹에서 호스트를 제거해도 제거 후에도 ID 보기가 호스트에 계속 할당됩니다. 제거된 호스트에서 ID 보기를 적용 취소하려면 **ipa idview-unapply id\_view\_name --hosts=name\_of\_the\_removed\_host** 명령을 실행해야 합니다.

이 섹션에서는 다음 목표를 달성하는 방법에 대해 설명합니다.

1.

호스트 그룹을 생성하고 여기에 호스트를 추가하는 방법.

2. 호스트 그룹에 ID 보기를 적용하는 방법.
3. 호스트 그룹에 새 호스트를 추가하고 ID 보기를 새 호스트에 적용하는 방법.

#### 사전 요구 사항

- 호스트 그룹에 적용할 ID 보기가 IdM에 있는지 확인합니다. 예를 들어 AD 사용자의 GID를 재정의하는 ID 보기를 생성하려면 ID 뷰를 사용하여 IdM 클라이언트의 AD 사용자에 대한 기본 신뢰 보기 속성 덮어쓰기를 참조하십시오.

#### 절차

1. 호스트 그룹을 생성하고 여기에 호스트를 추가합니다.
  - a. 호스트 그룹을 생성합니다. 예를 들어 이름이 **baltimore** 인 호스트 그룹을 생성하려면 다음을 수행합니다.
 

```
[root@server ~]# ipa hostgroup-add --desc="Baltimore hosts" baltimore
-----
Added hostgroup "baltimore"
-----
Host-group: baltimore
Description: Baltimore hosts
```
  - b. 호스트 그룹에 호스트를 추가합니다. 예를 들어 **host102** 및 **host103** 을 **baltimore** 호스트 그룹에 추가하려면 다음을 수행합니다.
 

```
[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. 호스트 그룹의 호스트에 ID 보기를 적용합니다. 예를 들어 **example\_for\_host1** ID 보기를 **baltimore** 호스트 그룹에 적용하려면 다음을 수행합니다.

```
[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.

새 호스트를 호스트 그룹에 추가하고 ID 보기를 새 호스트에 적용합니다.

a.

호스트 그룹에 새 호스트를 추가합니다. 예를 들어 **somehost.idm.example.com** 호스트를 **baltimore** 호스트 그룹에 추가하려면 다음을 수행합니다.

```
[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.

선택적으로 ID 보기 정보를 표시합니다. 예를 들어 **example\_for\_host1** ID 보기에 대한 세부 정보를 표시하려면 다음을 수행합니다.

```
[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
```

출력에서 ID 보기가 **baltimore** 호스트 그룹의 새로 추가된 호스트인 **somehost.idm.example.com**에 적용되지 않음을 보여줍니다.

c.

새 호스트에 ID 뷰를 적용합니다. 예를 들어 **example\_for\_host1** ID 보기를 **somehost.idm.example.com**에 적용하려면 다음을 수행합니다.

```
[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
-----
```

## 검증 단계

- ID 보기 정보를 다시 표시합니다.

```
[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
```

출력에 ID 보기가 **baltimore** 호스트 그룹에 새로 추가된 호스트인 **somehost.idm.example.com**에 적용됨을 보여줍니다.

## 28장. 수동으로 ID 범위 조정

IdM 서버는 고유한 사용자 ID(UID) 및 그룹 ID(GID) 번호를 생성합니다. 복제본에 다른 ID 범위를 생성하고 할당하면 동일한 ID 번호를 생성하지도 않습니다. 기본적으로 이 프로세스는 **Automatic**입니다. 그러나 IdM 서버 설치 중에 IdM ID 범위를 수동으로 조정하거나 복제본의 DNA ID 범위를 수동으로 정의할 수 있습니다.

### 28.1. ID 범위

ID 번호는 **ID 범위**로 나뉩니다. 개별 서버 및 복제본에 대해 별도의 숫자 범위를 유지하면 항목에 대해 발행된 ID 번호가 다른 서버 또는 복제본에서 이미 사용되었을 가능성이 없어집니다.

두 가지 유형의 ID 범위가 있습니다.

- 첫 번째 서버를 설치하는 동안 할당되는 **IdM ID 범위**. 이 범위는 생성된 후에는 수정할 수 없습니다. 그러나 원래 ID 범위 외에 새 IdM ID 범위를 생성할 수 있습니다. 자세한 내용은 [자동 ID 범위 할당 및 새 IdM ID 범위 추가](#)를 참조하십시오.
- 사용자가 수정할 수 있는 **DNA( *Distributed Numeric Assignment* ) ID 범위**입니다. 기존 IdM ID 범위 내에 일치해야 합니다. 자세한 내용은 [DNA ID 범위가 수동으로 할당을 참조하십시오](#).

또한 복제본에는 다음 **DNA ID 범위**가 할당될 수 있습니다. 복제본은 현재 범위에서 ID가 아닌 경우 다음 범위를 사용합니다. 복제본이 삭제되면 다음 범위는 자동으로 할당되지 않으며 **수동으로 할당해야** 합니다.

범위는 도메인의 백엔드 **389 Directory Server** 인스턴스의 일부로 **DNA 플러그인**에 의해 서버와 복제본 간에 업데이트 및 공유됩니다.

**DNA 범위** 정의는 다음 두 가지 속성으로 설정됩니다.

- 서버의 사용 가능한 다음 번호: **DNA 범위의 낮은 끝**
- 범위 크기: **DNA 범위의 ID 수**

초기 하단 범위는 플러그인 인스턴스 구성 중에 설정됩니다. 그 후 플러그인은 하단 값을 업데이트합니다. 사용 가능한 숫자를 범위로 분리하면 서버에서 서로 겹치지 않고 지속적으로 숫자를 할당할 수 있습니다.

## 28.2. 자동 ID 범위 할당

### IdM ID 범위

기본적으로 IdM ID 범위는 IdM 서버 설치 중에 자동으로 할당됩니다. `ipa-server-install` 명령은 총 10,000개의 가능한 범위에서 200,000개의 ID 범위를 임의로 선택하고 할당합니다. 이러한 방식으로 임의의 범위를 선택하면 향후 두 개의 별도의 IdM 도메인을 병합하기로 결정하는 경우 ID 충돌이 발생할 가능성이 크게 감소합니다.

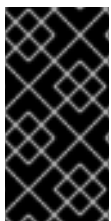


#### 참고

이 IdM ID 범위는 생성한 후에는 수정할 수 없습니다. DNA ID 범위 할당에 설명된 명령을 사용하여 DNA(Distributed Numeric Assignment) 범위만 수동으로 조정할 수 있습니다. IdM ID 범위와 일치하는 DNA 범위가 설치 중에 자동으로 생성됩니다.

### DNA ID 범위

단일 IdM 서버가 설치되어 있는 경우 전체 DNA ID 범위를 제어합니다. 새 복제본을 설치할 때 복제본은 자체 DNA ID 범위를 요청하면 서버 분할에 대한 초기 ID 범위가 서버 및 복제본 간에 분산됩니다. 복제본은 초기 서버에서 사용할 수 있는 나머지 DNA ID 범위의 절반을 수신합니다. 그런 다음 서버 및 복제본은 새 사용자 또는 그룹 항목에 대해 원래 ID 범위의 해당 부분을 사용합니다. 또한 복제본이 할당된 ID 범위를 고갈하고 100개 미만의 ID가 남아 있는 경우 복제본에서 사용 가능한 다른 서버에 연결하여 새 DNA ID 범위를 요청합니다.



#### 중요

복제본을 설치할 때 ID 범위를 즉시 수신하지 않습니다. 복제본은 예를 들어 사용자를 처음 추가할 때 DNA 플러그인이 사용될 때 ID 범위를 수신합니다.

복제본에서 DNA ID 범위를 요청하기 전에 초기 서버가 작동을 중지하면 복제본에서 ID 범위를 요청하도록 서버에 연결할 수 없습니다. 복제본에서 새 사용자를 추가하려고 하면 실패합니다. 이러한 상황에서는 비활성화된 서버에 할당된 ID 범위를 확인하고 복제본에 ID 범위를 수동으로 할당할 수 있습니다.

## 28.3. 서버 설치 중 IdM ID 범위 수동 할당

기본 동작을 재정의하고 무작위로 할당하는 대신 IdM ID 범위를 수동으로 설정할 수 있습니다.



### 중요

UID 값이 1000 이하인 ID 범위를 설정하지 마십시오. 이러한 값은 시스템 사용을 위해 예약되어 있습니다. 또한 0 값을 포함하는 ID 범위를 설정하지 마십시오. SSSD 서비스에서 0 ID 값을 처리하지 않습니다.

### 절차

- **ipa-server-install** 과 함께 다음 두 가지 옵션을 사용하여 서버 설치 중에 IdM ID 범위를 수동으로 정의할 수 있습니다.
  - **--idstart** 는 UID 및 GID 번호의 시작 값을 제공합니다.
  - **--idmax** 는 최대 UID 및 GID 번호를 제공합니다. 기본적으로 값은 **--idstart** 시작 값과 199,999입니다.

### 검증 단계

- ID 범위가 올바르게 할당되었는지 확인하려면 **ipa idrange-find** 명령을 사용하여 할당된 IdM ID 범위를 표시할 수 있습니다.

```
# 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
-----
```

## 28.4. 새 IdM ID 범위 추가

예를 들어 복제본이 ID가 부족하고 원래 IdM ID 범위가 고갈되는 경우와 같이 원래 IdM ID 범위 외에도 새 IdM ID 범위를 생성할 수 있습니다.



## 중요

새 IdM ID 범위를 추가해도 새 DNA ID 범위가 자동으로 생성되지 않습니다. 필요에 따라 복제본에 새 DNA ID 범위를 수동으로 할당해야 합니다. 이 작업을 수행하는 방법에 대한 자세한 내용은 [DNA ID 범위를 수동으로 할당](#)을 참조하십시오.

## 절차

1.

새 IdM ID 범위를 생성하려면 `ipa idrange-add` 명령을 사용합니다. 새 범위 이름, 범위의 첫 번째 ID 번호 및 범위 크기를 지정해야 합니다.

```
# 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.

선택 사항: ID 범위를 즉시 업데이트합니다.

a.

SSSD(System Security Services Daemon) 캐시를 지웁니다.

```
# sss_cache -E
```

b.

SSSD 데몬을 다시 시작합니다.

```
# systemctl restart sssd
```



## 참고

SSSD 캐시를 지우지 않고 서비스를 다시 시작하지 않으면 SSSD에서 도메인 목록 및 IdM 서버에 저장된 기타 구성 데이터를 업데이트할 때만 새 ID 범위를 탐지합니다.

## 검증 단계

•

`ipa idrange-find` 명령을 사용하여 새 범위가 올바르게 설정되었는지 확인할 수 있습니다.

```
# 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
-----
```

### 28.5. IDM ID 범위에서 보안 및 상대 식별자의 역할

IdM(Identity Management) ID 범위는 여러 매개변수로 정의됩니다.

- 범위 이름
- 범위의 첫 번째 **POSIX ID**
- 범위 크기: 범위의 **ID 수**
- 해당 **RID** 범위의 첫 번째 상대 식별자 (**RID**)
- 보조 **RID** 범위의 첫 번째 **RID**

`ipa idrange-show` 명령을 사용하여 이러한 값을 볼 수 있습니다.

```
$ 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
```

## 보안 식별자

로컬 도메인의 ID 범위 데이터는 IdM 서버에서 IdM 사용자 및 그룹에 고유한 보안 식별자 (SID)를 할당하는 데 사용됩니다. STS는 사용자 및 그룹 개체에 저장됩니다. 사용자의 SID는 다음으로 구성됩니다.

- 도메인 SID
- 사용자 상대 식별자 (RID)는 도메인 STS에 추가된 네 자리 32비트 값입니다.

예를 들어 도메인 SID가 S-1-5-21-123-456-789이고 이 도메인의 사용자 RID가 1008인 경우, 사용자에게 S-1-5-21-123-456-789-1008의 HEAD가 있습니다.

## 상대 식별자

RID 자체는 다음과 같은 방식으로 계산됩니다.

범위의 첫 번째 POSIX ID를 사용자의 POSIX UID에서 빼고 해당 RID 범위의 첫 번째 RID를 결과에 추가합니다. 예를 들어 *idmuser*의 UID가 196600008인 경우 첫 번째 POSIX ID는 *jenkinsfile600000*이고 첫 번째 RID는 1000이고 *idmuser*'s RID는 1008입니다.



### 참고

사용자의 RID 알고리즘을 계산하는 알고리즘은 지정된 POSIX ID가 해당 RID를 계산하기 전에 할당된 ID 범위에 속하는지 확인합니다. 예를 들어 첫 번째 ID가 *jenkinsfile600000*이고 범위 크기가 200000인 경우, 이 범위의 POSIX ID는 ID 범위 외부에 있고 알고리즘은 RID를 계산하지 않습니다.

## 보조 상대 식별자

IdM에서 POSIX UID는 POSIX GID와 같을 수 있습니다. 즉, *idmuser*가 196600008 UID가 있는 경우 GID가 196600008인 새 *idmgroup* 그룹을 계속 생성할 수 있습니다.

그러나 STS는 사용자 또는 그룹 한 개만 정의할 수 있습니다. *idmuser* 용으로 이미 생성된 S-1-5-21-123-456-789-1008의 SID는 *idmgroup*과 공유할 수 없습니다. *idmgroup*에 대해 대체 STS를 생성해야 합니다.

IdM은 보조 상대 식별자 (또는 보조 RID)를 사용하여 conflict replacess가 발생하지 않습니다. 이 보조 RID는 다음과 같이 구성됩니다.

- 2차 RID 기반
- 범위 크기(기본 범위 크기와 기본적으로 동일)

위의 예에서 보조 RID 베이스는 1000000으로 설정됩니다. 새로 생성된 *idmgroup*의 RID를 계산하려면 : 사용자의 POSIX UID에서 범위의 첫 번째 POSIX ID를 제거하고 보조 RID 범위의 첫 번째 RID 범위를 결과에 추가합니다. 따라서 *idmgroup*에는 RID 1000008이 할당됩니다. 그 결과 *idmgroup*의 SID는 S-1-5-21-123-456-789-1000008입니다.

IdM은 보조 RID를 사용하여 사용자 또는 그룹 오브젝트가 이전에 수동으로 설정된 POSIX ID를 사용하여 생성한 경우에만 STS를 계산합니다. 그렇지 않으면 자동 할당으로 동일한 ID를 두 번 할당하지 않습니다.

## 28.6. AD에 대한 신뢰를 제거한 후 ID 범위 제거

IdM과 AD(Active Directory) 환경 간의 신뢰를 제거한 경우 연결된 ID 범위를 제거할 수 있습니다.



### 주의

신뢰할 수 있는 도메인과 연결된 ID 범위에 할당된 ID는 IdM에 등록된 시스템의 파일 및 디렉터리 소유권에 계속 사용될 수 있습니다.

제거한 AD 신뢰에 해당하는 ID 범위를 제거하면 AD 사용자가 소유한 모든 파일 및 디렉터리의 소유권을 해결할 수 없습니다.

## 사전 요구 사항

- AD 환경에 대한 신뢰를 제거했습니다.

## 절차

1. 현재 사용 중인 모든 ID 범위를 표시합니다.

```
[root@server ~]# ipa idrange-find
```

2. 제거한 신뢰와 연결된 ID 범위의 이름을 확인합니다. ID 범위 이름의 첫 번째 부분은 신뢰 이름입니다(예: **AD.EXAMPLE.COM\_id\_range**).

3. 범위를 제거합니다.

```
[root@server ~]# ipa idrange-del AD.EXAMPLE.COM_id_range
```

4. SSSD 서비스를 다시 시작하여 제거한 ID 범위에 대한 참조를 제거합니다.

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

#### 추가 리소스

- [명령줄을 사용하여 신뢰 제거를 참조하십시오.](#)
- [IdM 웹 UI를 사용하여 신뢰 제거를 참조하십시오.](#)

## 28.7. 현재 할당된 DNA ID 범위 표시

서버에 현재 활성화된 DNA(Distributed Numeric Assignment) ID 범위 및 할당된 경우 다음 DNA 범위를 모두 표시할 수 있습니다.

#### 절차

- 토폴로지의 서버에 대해 구성된 DNA ID 범위를 표시하려면 다음 명령을 사용합니다.
  - **ipa-replica-manage dnarange-show** 는 지정된 서버에서만 서버를 지정하는 경우 또는 지정된 서버에서만 서버를 지정하는 경우 현재 DNA ID 범위를 표시합니다. 예를 들면 다음과 같습니다.

```
# 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 dnarange-show** 는 현재 모든 서버에 설정된 다음 **DNA ID** 범위를 표시하거나, 서버를 지정하는 경우 서버를 지정된 서버에서만 표시합니다. 예를 들면 다음과 같습니다.

```
# ipa-replica-manage dnarange-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 dnarange-show serverA.example.com
serverA.example.com: 2001-2500
```

## 28.8. 수동 ID 범위 할당

특정 상황에서는 다음과 같이 **DNA(Distributed Numeric Assignment) ID** 범위를 수동으로 할당해야 합니다.

●

복제본에 ID가 부족하여 **IdM ID** 범위가 고갈됨

복제본에서 할당된 **DNA ID** 범위를 소진했으며, **IdM** 범위에서 사용 가능한 ID가 더 이상 없기 때문에 추가 ID를 요청하는 데 실패했습니다.

이 상황을 해결하려면 복제본에 할당된 **DNA ID** 범위를 확장합니다. 이 작업은 다음 두 가지 방법으로 수행할 수 있습니다.

○

다른 복제본에 할당된 **DNA ID** 범위를 단축한 다음, 새로 사용 가능한 값을 고갈된 복제본에 할당합니다.

○

새 **IdM ID** 범위를 생성한 다음 생성된 이 **IdM** 범위 내의 복제본에 대한 새 **DNA ID** 범위를 설정합니다.

새 **IdM ID** 범위를 생성하는 방법에 대한 자세한 내용은 [새 IdM ID 범위 추가](#)를 참조하십시오.

## •

복제본이 중지되지 않은 기능

복제본의 **DNA ID** 범위는 복제본이 작동하지 않고 삭제해야 할 때 자동으로 검색되지 않으므로 복제본에 이전에 할당된 **DNA ID** 범위를 사용할 수 없게 됩니다. **DNA ID** 범위를 복구하고 다른 복제본에서 사용할 수 있도록 하고 싶습니다.

이렇게 하려면 해당 범위를 다른 서버에 수동으로 할당하기 전에 **ID 범위 값이 무엇인지 확인합니다**. 또한 **UID** 또는 **GID**가 중복되지 않도록 하려면 복구된 범위의 **ID** 값이 이전에 사용자 또는 그룹에 할당되어 있지 않은지 확인합니다. 기존 사용자 및 그룹의 **UID** 및 **GID**를 검사하여 이 작업을 수행할 수 있습니다.

**DNA ID** 범위를 수동으로 할당하는 명령을 사용하여 **DNA ID** 범위를 복제본에 수동으로 할당할 수 있습니다.



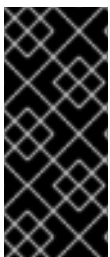
참고

새 **DNA ID** 범위를 할당하는 경우 서버 또는 복제본에 이미 존재하는 항목의 **UID**는 동일하게 유지됩니다. 현재 **DNA ID** 범위를 변경하더라도 **IdM**은 과거에 할당된 범위의 레코드를 유지하므로 문제가 발생하지 않습니다.

## 28.9. 수동으로 DNA ID 범위 할당

경우에 따라 기존 복제본에 **DNA(Distributed Numeric Assignment) ID** 범위를 수동으로 할당하여 작동하지 않는 복제본에 할당된 **DNA ID** 범위를 다시 할당해야 할 수 있습니다. 자세한 내용은 [수동 ID 범위 할당](#) 을 참조하십시오.

**DNA ID** 범위를 수동으로 조정하는 경우, 새로 조정된 범위가 **IdM ID** 범위에 포함되어 있는지 확인하십시오. **ipa idrange-find** 명령을 사용하여 확인할 수 있습니다. 그렇지 않으면 명령이 실패합니다.



중요

겹치는 **ID** 범위를 만들지 않도록 주의하십시오. 서버 또는 복제본에 할당하는 **ID** 범위 중 하나라도 겹치면 다른 항목에 동일한 **ID** 값을 할당하는 두 개의 다른 서버가 발생할 수 있습니다.

## 사전 요구 사항

## •

**선택 사항:** 기능이 아닌 복제본에서 **DNA ID** 범위를 복구하는 경우 먼저 현재 할당된 **DNA ID** 범위를 표시하는 데 설명된 명령을 사용하여 **ID** 범위를 찾습니다.

## 절차

- 지정된 서버에 대한 현재 **DNA ID** 범위를 정의하려면 **ipa-replica-manage dnarange-set** 을 사용합니다.

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

- 지정된 서버에 대한 다음 **DNA ID** 범위를 정의하려면 **ipa-replica-manage dnanextrange-set** 를 사용합니다.

```
# ipa-replica-manage dnanextrange-set serverB.example.com 1500-5000
```

## 검증 단계

- 현재 할당된 **DNA ID** 범위를 표시하는 명령을 사용하여 새 **DNA** 범위가 올바르게 설정되었는지 확인할 수 있습니다.

## 29장. 하위 ID 범위 수동 관리

컨테이너화된 환경에서는 IdM 사용자가 하위 ID 범위를 수동으로 할당해야 하는 경우가 있습니다. 다음 지침은 **subID** 범위를 관리하는 데 도움이 됩니다.

### 29.1. IDM CLI를 사용하여 SUBID 범위 생성

하위 ID 범위를 생성하여 수동으로 사용자에게 할당할 수 있습니다. 사용자 이름 *jsmith*가 ipa 서버에 있다고 가정합니다.

#### 사전 요구 사항

- IdM 사용자가 있습니다.
- 유효한 Kerberos 티켓을 받습니다. 자세한 내용은 [웹 UI에서 IdM으로 로그인: Kerberos 티켓 사용](#)을 참조하십시오.
- 루트 권한.

#### 절차

1. 기존 하위 ID 범위를 확인합니다.

```
# ipa subid-find
```

2. subID 범위가 없는 경우 다음 명령을 입력하여 새 하위 ID 범위를 사용자에게 생성하고 할당합니다.

```
# ipa subid-generate --owner=jsmith
```

```
Added subordinate id "359dfcef-6b76-4911-bd37-bb5b66b8c418"
```

```
Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
```

```
Description: auto-assigned subid
```

```
Owner: jsmith
```

```
SubUID range start: 2147483648
```

```
SubUID range size: 65536
```

```
SubGID range start: 2147483648
```

```
SubGID range size: 65536
```

3.

또는 모든 사용자에게 새 하위 ID 범위를 생성하고 할당합니다.

```
# /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
```

기본적으로 새 IdM 사용자에게 하위 ID 범위를 할당하려면 다음 옵션을 활성화합니다.

```
# ipa config-mod --user-default-subid=True
```

#### 검증

1.

사용자에게 하위 ID 범위가 할당되어 있는지 확인하려면 다음 명령을 입력합니다.

```
# ipa subid-find --owner=jsmith

1 subordinate id matched

Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
Owner: jsmith
SubUID range start: 2147483648
SubUID range size: 65536
SubGID range start: 2147483648
SubGID range size: 65536

Number of entries returned 1
```

## 29.2. IDM WEBUI 인터페이스를 사용하여 하위 ID 범위 생성

하위 ID 범위를 생성하여 IdM WebUI 인터페이스의 사용자에게 할당할 수 있습니다.

#### 사전 요구 사항

- IdM 사용자가 있습니다.
- 유효한 Kerberos 티켓을 받습니다. 자세한 내용은 [웹 UI에서 IdM으로 로그인: Kerberos 티켓 사용](#)을 참조하십시오.

- 루트 권한.

#### 절차

1. **IdM WebUI** 인터페이스에서는 **ID** 탭을 확장하고 **ID** 대체 옵션을 선택합니다.
2. **Subordinate ID** 인터페이스가 표시되면 인터페이스 오른쪽 상단에 있는 **Add** (추가) 버튼을 클릭합니다. **"Add subid"** 창이 표시됩니다.
3. **"Add subid"** 창에서 **subID** 범위를 할당하려는 사용자인 소유자를 선택합니다.
4. 추가 버튼을 클릭합니다.

#### 검증

1. **Subordinate ID** 탭에서 테이블을 확인합니다. 새 레코드가 표시되고 소유자는 하위 **ID** 범위를 할당하는 사용자입니다.

### 29.3. IDM CLI를 사용하여 기존 SUBID 범위 관리

**subID** 범위를 검색하고 필요한 경우 특정 하나에 대한 정보를 표시할 수 있습니다. 사용자 이름 *jsmith*가 *ipa* 서버에 있다고 가정합니다.

#### 사전 요구 사항

- **IdM** 사용자가 있습니다.

#### 절차

1. 고유 **ID** 해시를 알고 있는 경우 **subID** 범위에 대한 세부 정보를 표시하려면 다음 명령을 입력합니다.

```
# ipa subid-show 359dfcef-6b76-4911-bd37-bb5b66b8c418
```

```
Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
Owner: jsmith
SubUID range start: 2147483648
```

```
SubUID range size: 65536
SubGID range start: 2147483648
SubGID range size: 65536
```

2.

해당 범위의 하위 ID가 있는 경우 하위 ID 범위에 대한 세부 정보를 찾으려면 다음 명령을 사용합니다.

```
# ipa subid-match --subuid=2147483648

1 subordinate id matched

Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
Owner: uid=jsmith
SubUID range start: 2147483648
SubUID range size: 65536
SubGID range start: 2147483648
SubGID range size: 65536

Number of entries returned 1
```

#### 29.4. GETSUBID 명령을 사용하여 하위 ID 범위 나열

하위 ID 범위(예: IdM 환경의 **user 1**)를 나열하려면 아래 지침을 따르십시오.

##### 사전 요구 사항

- **user1** 은 IdM에 있습니다.
- **shadow-utils-subid** 패키지가 설치됩니다.

##### 절차

1.

**subid: sss** 레코드를 **/etc/nsswitch.conf** 파일에 포함합니다.

하위 필드에 대해 하나의 값만 제공할 수 있습니다. **subid** 필드를 **sss** 값으로 설정하면 **utils** 에 IdM 설정에서 **subID** 범위를 사용하도록 지시합니다. **file** 값 또는 **no** 값은 **/etc/subuid** 및 **/etc/subgid** 파일의 하위 ID 범위를 사용하도록 **utils**를 설정합니다.

2.

사용자의 하위 ID 범위를 나열합니다.

```
# getsubids user1  
0: user1 2147483648 65536
```

## 30장. IDM CLI에서 호스트 관리

이 장에서는 **IdM(Identity Management)**의 [호스트 및 호스트 항목](#) 과 **IdM CLI**의 호스트 및 호스트 항목을 관리할 때 수행되는 다음 작업을 소개합니다.

- [호스트 등록](#)
- [IdM 호스트 항목 추가](#)
- [IdM 호스트 항목 삭제](#)
- [다시 등록 호스트](#)
- [호스트 이름 변경](#)
- [호스트 비활성화](#)
- [호스트 다시 활성화](#)

이 장에는 사전 요구 사항, 컨텍스트 및 이러한 작업의 결과에 대한 [개요 테이블](#) 도 포함되어 있습니다.

### 30.1. IDM의 호스트

**IdM(Identity Management)**은 이러한 ID를 관리합니다.

- 사용자
- 서비스
- 호스트

호스트는 시스템을 나타냅니다. **IdM ID**로, 호스트에 **IdM LDAP**에 대한 항목이 있으며, 이는 **IdM 서버**의 **389 Directory Server** 인스턴스입니다.

**IdM LDAP**의 호스트 항목은 다른 호스트와 도메인 내의 서비스 간 관계를 설정하는 데 사용됩니다. 이러한 관계는 도메인 내의 호스트에 대한 권한 부여 및 제어를 *위임*하는 과정의 일부입니다. 모든 호스트는 호스트 기반 액세스 제어 (**HBAC**) 규칙에 사용할 수 있습니다.

**IdM 도메인**은 일반 **ID** 정보, 공통 정책 및 공유 서비스를 통해 시스템 간의 공통성을 설정합니다. 도메인 클라이언트에 속하는 모든 시스템은 도메인의 클라이언트로 기능하므로 도메인에서 제공하는 서비스를 사용합니다. **IdM 도메인**은 특히 시스템용 세 가지 주요 서비스를 제공합니다.

- **DNS**
- **Kerberos**
- 인증서 관리

**IdM**의 호스트는 해당 호스트에서 실행되는 서비스와 밀접하게 연결됩니다.

- 서비스 항목은 호스트와 연결됩니다.
- 호스트는 호스트와 서비스 **Kerberos** 사용자를 모두 저장합니다.

## 30.2. 호스트 등록

이 섹션에서는 **IdM** 클라이언트로 호스트 등록 및 등록 후 수행되는 작업을 설명합니다. 섹션은 **IdM** 호스트 등록과 **IdM** 사용자를 비교합니다. 이 섹션에서는 호스트에서 사용할 수 있는 대체 인증 유형도 간략하게 설명합니다.

호스트 등록은 다음으로 구성됩니다.

- **IdM LDAP**에 호스트 항목 생성: **IdM CLI**에서 **ipa host-add 명령** 또는 이에 해당하는 **IdM 웹 UI 작업**을 사용할 수 있습니다.

- 호스트에서 **IdM** 서비스 구성(예: **SSSD(System Security Services Daemon)**, **Kerberos**, **certmonger** 및 **host**를 **IdM** 도메인에 가입합니다.

두 작업은 개별적으로 또는 함께 수행할 수 있습니다.

별로도 수행하는 경우 서로 다른 수준의 권한을 가진 두 사용자 간에 두 작업을 분할할 수 있습니다. 이는 대규모 배포에 유용합니다.

**ipa-client-install** 명령은 두 가지 작업을 함께 수행할 수 있습니다. 이 명령은 해당 항목이 아직 없는 경우 **IdM LDAP**에 호스트 항목을 생성하고 호스트에 대한 **Kerberos** 및 **SSSD** 서비스 모두를 구성합니다. 이 명령은 **IdM** 도메인 내의 호스트를 가져와서 연결할 **IdM** 서버를 식별할 수 있습니다. 호스트가 **IdM**에서 관리하는 **DNS** 영역에 속하는 경우 **ipa-client-install** 은 호스트에 대한 **DNS** 레코드도 추가합니다. 명령은 클라이언트에서 실행해야 합니다.

### 30.3. 호스트 등록에 필요한 사용자 권한

호스트 등록 작업을 수행하려면 권한이 없는 사용자가 **IdM** 도메인에 원치 않는 시스템을 추가하지 못하도록 인증을 수행해야 합니다. 필요한 권한은 다음과 같은 여러 요인에 따라 달라집니다.

- 호스트 항목이 **ipa-client-install** 실행과 별도로 생성되는 경우
- 등록에 일회성 암호(OTP)를 사용하는 경우

**IdM LDAP**에서 호스트 항목을 수동으로 생성하는 선택 사항의 사용자 권한

**ipa host-add CLI** 명령 또는 **IdM 웹 UI**를 사용하여 **IdM LDAP**에 호스트 항목을 생성하는 데 필요한 사용자 권한은 호스트 관리자입니다. 호스트 관리자 권한은 **IT professionals** 역할을 통해 얻을 수 있습니다.

클라이언트를 **IdM** 도메인에 가입하기 위한 사용자 권한

호스트는 **ipa-client-install** 명령을 실행하는 동안 **IdM** 클라이언트로 구성됩니다. **ipa-client-install** 명령을 실행하는 데 필요한 인증 정보 수준은 다음 중 사용자가 직접 찾을 수 있는 시나리오에 따라 다릅니다.

- **IdM LDAP**의 호스트 항목이 없습니다. 이 시나리오에서는 전체 관리자의 자격 증명 또는 호스

트 관리자 역할이 필요합니다. 전체 관리자는 **admins** 그룹의 멤버입니다. 호스트 관리자 역할은 호스트를 추가하고 호스트를 등록할 수 있는 권한을 제공합니다. 이 시나리오에 대한 자세한 내용은 [사용자 자격 증명을 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치.](#)

- 

**IdM LDAP**의 호스트 항목이 있습니다. 이 시나리오에서는 **ipa-client-install** 을 성공적으로 실행하려면 제한된 관리자의 인증 정보가 필요합니다. 이 경우 제한된 관리자에게는 **Enrollment Administrator** 역할이 있으며 이는 호스트 등록 권한을 제공합니다. 자세한 내용은 [사용자 자격 증명을 사용하여 클라이언트 설치: 대화식 설치.](#)

- 

**IdM LDAP**의 호스트 항목이 있으며 전체 또는 제한된 관리자가 호스트에 대한 **OTP**가 생성되었습니다. 이 시나리오에서는 **ipa-client-install** 명령을 **--password** 옵션과 함께 실행하여 올바른 **OTP**를 제공하는 경우 일반 사용자로 **IdM** 클라이언트를 설치할 수 있습니다. 자세한 내용은 [일회성 암호: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오.](#)

등록 후 **IdM** 호스트는 모든 새 세션을 인증하여 **IdM** 리소스에 액세스할 수 있습니다. **IdM** 서버가 시스템을 신뢰하고 해당 시스템에 설치된 클라이언트 소프트웨어에서 **IdM** 연결을 수락하려면 시스템 인증이 필요합니다. 클라이언트를 인증한 후 **IdM** 서버는 해당 요청에 응답할 수 있습니다.

#### 30.4. IDM 호스트 및 사용자의 등록 및 인증: 비교

**IdM**의 사용자와 호스트 간에는 많은 유사성이 있습니다. 이 섹션에서는 등록 단계에서 확인할 수 있는 유사성과 배포 단계에서 인증과 관련된 몇 가지에 대해 설명합니다.

- 

등록 단계([사용자 및 호스트 등록](#)):

- 

관리자는 사용자 또는 호스트가 실제로 **IdM**에 참여하기 전에 사용자와 호스트에 대한 **LDAP** 항목을 생성할 수 있습니다. **stage** 사용자의 경우 명령은 **ipa stageuser-add** 입니다. **host**는 **ipa host-add** 입니다.

- 

사용자 암호를 어느 정도까지 줄일 수 있는 **키 테이블** 또는 축약, 키맵, 대칭 키가 포함된 파일은 호스트에서 **ipa-client-install** 명령을 실행하는 동안 생성됩니다. 이로 인해 호스트가 **IdM** 영역에 결합됩니다. **Analogically**는 사용자가 계정을 활성화할 때 암호를 생성해야 하므로 **IdM** 영역에 가입해야 합니다.

- 

사용자 암호는 사용자의 기본 인증 방법이지만 **keytab**은 호스트의 기본 인증 방법입니다. **keytab**은 호스트의 파일에 저장됩니다.

표 30.1. 사용자 및 호스트 등록

| 동작     | 사용자                                               | 호스트                                                     |
|--------|---------------------------------------------------|---------------------------------------------------------|
| 사전 등록  | \$ ipa stageuser-add <i>user_name</i> [-password] | \$ ipa host-add <i>host_name</i> [--random]             |
| 계정 활성화 | \$ ipa stageuser-activate <i>user_name</i>        | \$ IPA-client install [--password]<br>(호스트 자체에서 실행해야 함) |

- 배포 단계(사용자 및 호스트 세션 인증)
  - 사용자가 새 세션을 시작하면 사용자는 암호를 사용하여 인증합니다. 마찬가지로, 호스트가 키탭 파일을 표시하여 인증합니다. SSSD(System Security Services Daemon)는 백그라운드에서 이 프로세스를 관리합니다.
  - 인증에 성공하면 사용자 또는 호스트는 TGT(Kerberos 티켓 부여 티켓)를 가져옵니다.
  - TGT는 특정 서비스에 대한 특정 티켓을 얻는 데 사용됩니다.

표 30.2. 사용자 및 호스트 세션 인증

|               | 사용자                                  | 호스트                                  |
|---------------|--------------------------------------|--------------------------------------|
| 기본 인증 수단      | 암호                                   | keytabs                              |
| 세션 시작(최신 사용자) | \$ kinit <i>user_name</i>            | [Winding on the host]                |
| 성공적인 인증의 결과   | TGT 는 특정 서비스에 대한 액세스 권한을 얻는 데 사용됩니다. | TGT 는 특정 서비스에 대한 액세스 권한을 얻는 데 사용됩니다. |

TGT 및 기타 Kerberos 티켓은 서버에서 정의한 Kerberos 서비스 및 정책의 일부로 생성됩니다. Kerberos 티켓의 초기 부여, Kerberos 자격 증명 갱신, Kerberos 세션 삭제도 IdM 서비스에서 자동으로 처리됩니다.

### IdM 호스트에 대한 대체 인증 옵션

IdM은 키탭 외에도 두 가지 유형의 시스템 인증을 지원합니다.

- SSH 키. 호스트의 SSH 공개 키가 생성되어 호스트 항목에 업로드됩니다. 여기에서 SSSD(System Security Services Daemon)는 IdM을 ID 공급자로 사용하며 OpenSSH 및 기타

서비스와 협력하여 IdM에 있는 공개 키를 참조할 수 있습니다.

•

시스템 인증서. 이 경우 시스템은 IdM 서버의 인증 기관에서 발행한 SSL 인증서를 사용한 다음 IdM 디렉터리 서버에 저장됩니다. 그런 다음 인증서를 서버에 인증할 때 존재하는 시스템으로 전송됩니다. 클라이언트에서 인증서는 **certmonger** 라는 서비스로 관리됩니다.

### 30.5. 호스트 작업

이 섹션에는 호스트 등록 및 활성화와 관련된 가장 일반적인 작업을 나열하고 사전 요구 사항, 컨텍스트 및 수행의 결과에 대해 설명합니다.

표 30.3. 호스트 운영 부분 1

| 동작         | 동작의 전제 조건은 무엇입니까?                                                      | 언제 명령을 실행하는 것이 적합합니까?                                | 시스템 관리자가 작업을 수행하는 방법은 무엇입니까? 실행 중인 명령은 무엇입니까?                                                                                                                                                                                                                                                                                                                  |
|------------|------------------------------------------------------------------------|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 클라이언트 등록   | <i>Installing Identity Management</i> 의 ID 관리 클라이언트 설치 시스템을 준비/참조하십시오. | 호스트가 IdM 영역에 참여하도록 하려면 다음을 수행합니다.                    | IdM 도메인에 있는 클라이언트로 시스템을 등록하는 것은 두 부분으로 구성된 프로세스입니다. <b>ipa host-add</b> 명령을 실행할 때 클라이언트(및 389 Directory Server 인스턴스에 저장)에 대한 호스트 항목이 생성되고, 클라이언트를 프로비저닝하기 위해 keytab이 생성됩니다. 두 부분 모두 <b>ipa-client-install</b> 명령을 통해 자동으로 수행됩니다. 또한 이러한 단계를 별도로 수행할 수 있습니다. 이를 통해 관리자는 실제로 클라이언트를 구성하기 전에 시스템과 IdM을 준비할 수 있습니다. 이를 통해 대규모 배포를 포함하여 보다 유연한 설정 시나리오를 사용할 수 있습니다. |
| 클라이언트 비활성화 | 호스트에 IdM에 항목이 있어야 합니다. 호스트에는 활성 keytab이 있어야 합니다.                       | 유지 관리를 위해 IdM 영역에서 호스트를 임시로 제거하려는 경우 다음을 수행할 수 있습니다. | <b>ipa host-disable host_name</b>                                                                                                                                                                                                                                                                                                                              |
| 클라이언트 활성화  | 호스트에 IdM에 항목이 있어야 합니다.                                                 | 일시적으로 비활성화된 호스트가 다시 활성화되도록 하려는 경우                    | <b>ipa-getkeytab</b>                                                                                                                                                                                                                                                                                                                                           |

| 동작          | 동작의 전체 조건은 무엇입니까?         | 언제 명령을 실행하는 것이 적합합니까?                      | 시스템 관리자가 작업을 수행하는 방법은 무엇입니까? 실행 중인 명령은 무엇입니까?                         |
|-------------|---------------------------|--------------------------------------------|-----------------------------------------------------------------------|
| 클라이언트 다시 등록 | 호스트에 IdM에 en 항목이 있어야 합니다. | 원래 호스트가 손실되었지만 동일한 호스트 이름을 가진 호스트를 설치했습니다. | <b>ipa-client-install --keytab or ipa-client-install --force-join</b> |
| 고객 등록 취소    | 호스트에 IdM에 항목이 있어야 합니다.    | IdM 영역에서 호스트를 영구적으로 제거하려는 경우.              | <b>ipa-client-install --uninstall</b>                                 |

표 30.4. 호스트 운영 부분 2

| 동작          | 관리자가 명령을 실행할 수 있는 시스템은 무엇입니까?                                                                                          | 작업이 수행되면 어떻게 됩니까? IdM에서 호스트가 작동하는 경우의 결과는 무엇입니까? 어떤 제한 사항이 도입/제거됩니까?                                                                                                    |
|-------------|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 클라이언트 등록    | 2단계 등록의 경우: <b>ipa host-add</b> 를 모든 IdM 클라이언트에서 실행할 수 있습니다. <b>ipa-client-install</b> 의 두 번째 단계는 클라이언트 자체에서 실행해야 합니다. | 기본적으로 인증 및 권한 부여를 위해 IdM 서버에 연결하도록 SSSD를 구성합니다. 선택적으로 Kerberos 및 LDAP를 통한 IdM 서버와 작동하도록 Pluggable Authentication Module(PAM) 및 Name Switching Service(NSS)를 구성할 수 있습니다. |
| 클라이언트 비활성화  | IdM의 모든 시스템, 호스트 자체도                                                                                                   | 호스트의 Kerberos 키 및 SSL 인증서가 무효화되고 호스트에서 실행 중인 모든 서비스가 비활성화됩니다.                                                                                                           |
| 클라이언트 활성화   | IdM에 있는 모든 시스템. 비활성화된 호스트에서 실행되는 경우 LDAP 인증 정보를 제공해야 합니다.                                                              | 호스트의 Kerberos 키와 SSL 인증서가 다시 유효하며 호스트에서 실행 중인 모든 IdM 서비스가 다시 활성화됩니다.                                                                                                    |
| 클라이언트 다시 등록 | 다시 등록할 호스트입니다. LDAP 인증 정보를 제공해야 합니다.                                                                                   | 호스트에 대해 새 Kerberos 키가 생성되어 이전 키를 대체합니다.                                                                                                                                 |

| 동작       | 관리자가 명령을 실행할 수 있는 시스템은 무엇입니까? | 작업이 수행되면 어떻게 됩니까? IdM에서 호스트가 작동하는 경우의 결과는 무엇입니까? 어떤 제한 사항이 도입/제거됩니까?                                                                                                                                                                                                                                                                   |
|----------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 고객 등록 취소 | 호스트가 등록되지 않음.                 | 이 명령은 IdM을 구성 해제하고 시스템을 이전 상태로 되돌립니다. 이 프로세스의 일부는 IdM 서버에서 호스트를 해제하는 것입니다. Unenrollment은 IdM 서버에서 주 키를 비활성화하는 것으로 구성됩니다. <code>/etc/krb5.keytab</code> (호스트/ <code>&lt;fqdn&gt;@REALM</code> )의 시스템 주체는 IdM 서버를 인증하는 데 사용됩니다. 이 주체가 존재하지 않으면 unenrollment이 실패하고 관리자가 호스트 주체( <code>ipa host-disable &lt;fqdn&gt;</code> )를 비활성화해야 합니다. |

### 30.6. IDM LDAP의 호스트 항목

이 섹션에서는 IdM(Identity Management)의 호스트 항목과 포함할 수 있는 특성에 대해 설명합니다.

LDAP 호스트 항목에는 IdM 내의 클라이언트에 대한 모든 관련 정보가 포함되어 있습니다.

- 호스트와 관련된 서비스 항목
- 호스트 및 서비스 주체
- 액세스 제어 규칙
- 물리적 위치 및 운영 체제와 같은 머신 정보



#### 참고

IdM 웹 UI ID → 호스트 탭에 IdM LDAP에 저장된 특정 호스트에 대한 모든 정보가 표시되지 않습니다.

#### 호스트 항목 구성 속성

호스트 항목은 물리적 위치, MAC 주소, 키 및 인증서와 같이 시스템 구성 외부에 있는 호스트에 대한 정보를 포함할 수 있습니다.

이 정보는 수동으로 생성되는 경우 호스트 항목을 생성할 때 설정할 수 있습니다. 또는 이 정보의 대부분

분은 호스트가 도메인에 등록된 후 호스트 항목에 추가할 수 있습니다.

표 30.5. 호스트 구성 속성

| UI 필드               | 명령줄 옵션                                    | 설명                                                                                                                                               |
|---------------------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 설명                  | <b>--desc</b> = <i>description</i>        | 호스트에 대한 설명입니다.                                                                                                                                   |
| 지역성                 | <b>--locality</b> = <i>locality</i>       | 호스트의 지리적 위치입니다.                                                                                                                                  |
| 위치                  | <b>--location</b> = <i>location</i>       | 데이터 센터 랙과 같은 호스트의 물리적 위치.                                                                                                                        |
| 플랫폼                 | <b>--platform</b> = <i>string</i>         | 호스트 하드웨어 또는 아키텍처.                                                                                                                                |
| 운영 체제               | <b>--os</b> = <i>string</i>               | 호스트의 운영 체제 및 버전입니다.                                                                                                                              |
| MAC 주소              | <b>--macaddress</b> = <i>address</i>      | 호스트의 MAC 주소입니다. 이는 다중 값 특성입니다. MAC 주소는 NIS 플러그인에서 호스트에 대한 NIS 이더넷 맵을 생성하는 데 사용됩니다.                                                               |
| SSH 공개 키            | <b>--sshpkey</b> = <i>string</i>          | 호스트의 전체 SSH 공개 키입니다. 이는 다중 값 특성이므로 여러 키를 설정할 수 있습니다.                                                                                             |
| 보안 주체 이름( 편집할 수 없음) | <b>--principalname</b> = <i>principal</i> | 호스트에 대한 Kerberos 사용자 이름입니다. 다른 보안 주체가 <b>-p</b> 에 명시적으로 설정되어 있지 않은 경우 클라이언트 설치 중에 기본적으로 호스트 이름으로 설정됩니다. 명령줄 도구를 사용하여 변경할 수 있지만 UI에서는 변경할 수 없습니다. |
| 일회성 암호 설정           | <b>--password</b> = <i>string</i>         | 이 옵션은 대규모 등록에 사용할 수 있는 호스트의 암호를 설정합니다.                                                                                                           |
| -                   | <b>--random</b>                           | 이 옵션은 대규모 등록에 사용할 임의의 암호를 생성합니다.                                                                                                                 |
| -                   | <b>--certificate</b> = <i>string</i>      | 호스트에 대한 인증서 Blob입니다.                                                                                                                             |
| -                   | <b>--updatedns</b>                        | 이렇게 하면 IP 주소가 변경되면 호스트가 DNS 항목을 동적으로 업데이트할 수 있는지 여부가 설정됩니다.                                                                                      |

### 30.7. IDM CLI에서 IDM 호스트 항목 추가

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 IdM(Identity Management)에 호스트 항목을 추가하는 방법을 설명합니다.

호스트 항목은 **host-add** 명령을 사용하여 생성됩니다. 이 명령을 수행하면 IdM 디렉터리 서버에 호스트 항목이 추가됩니다. CLI에서 ipa 도움말 호스트 를 입력하여 ipa 호스트 manpage를 참조하여 **host-add** 에서 사용 가능한 전체 옵션 목록을 가져옵니다.

IdM에 호스트를 추가할 때 몇 가지 시나리오가 있습니다.

- 기본적으로 **Kerberos** 영역에 클라이언트를 추가하고 IdM LDAP 서버에 항목을 생성하려면 클라이언트 호스트 이름만 지정합니다.

```
$ ipa host-add client1.example.com
```

- DNS를 관리하도록 IdM 서버가 구성된 경우 **--ip-address** 옵션을 사용하여 호스트를 DNS 리소스 레코드에 추가합니다.

예 30.1. 고정 IP 주소를 사용하여 호스트 항목 생성

```
$ ipa host-add --ip-address=192.168.166.31 client1.example.com
```

- 추가할 호스트에 고정 IP 주소가 없거나 클라이언트가 구성된 시점에 IP 주소를 알 수 없는 경우 **ipa host-add** 명령과 함께 **--force** 옵션을 사용합니다.

예 30.2. DHCP를 사용하여 호스트 항목 생성

```
$ ipa host-add --force client1.example.com
```

예를 들어 랩탑은 IdM 클라이언트로 사전 구성될 수 있지만, 구성된 시점에는 IP 주소가 없습니다. **--force** 를 사용하면 기본적으로 IdM DNS 서비스에 자리 표시자 항목을 생성합니다. DNS 서비스가 레코드를 동적으로 업데이트하면 호스트의 현재 IP 주소가 감지되고 해당 DNS 레코드가 업데이트됩니다.

### 30.8. IDm CLI에서 호스트 항목 삭제

- **host-del** 명령을 사용하여 호스트 레코드를 삭제합니다. IdM 도메인에 DNS가 통합된 경우 **--updatedns** 옵션을 사용하여 DNS에서 호스트에 대한 모든 종류의 관련 레코드를 제거합니다.

```
$ ipa host-del --updatedns client1.example.com
```

### 30.9. IDENTITY MANAGEMENT 클라이언트 다시 등록

이 섹션에서는 ID 관리 클라이언트를 다시 등록할 수 있는 다양한 방법에 대해 설명합니다.

#### 30.9.1. IdM의 클라이언트 re-enrollment

이 섹션에서는 IdM(Identity Management) 클라이언트를 다시 설정하는 방법에 대해 설명합니다.

클라이언트 시스템이 제거되고 IdM 서버와의 연결이 손실된 경우(예: 클라이언트의 하드웨어 오류로 인해) 키를 계속 사용하면 클라이언트를 다시 인증할 수 있습니다. 이 시나리오에서는 클라이언트가 동일한 호스트 이름을 가진 IdM 환경에서 다시 가져오려고 합니다.

재등록 중에 클라이언트는 새로운 Kerberos 키와 SSH 키를 생성하지만 LDAP 데이터베이스의 클라이언트 ID는 변경되지 않습니다. 재부팅 후 시스템은 IdM 서버와의 연결이 손실되기 전에 이전과 동일한 FQDN 을 사용하는 것과 동일한 LDAP 오브젝트에 키 및 기타 정보가 있습니다.

#### 중요

도메인 항목이 아직 활성화된 클라이언트만 다시 등록할 수 있습니다. 클라이언트 설치 제거( `ipa-client-install --uninstall` 사용) 또는 호스트 항목을 비활성화한 경우( `ipa host-disable` 사용) 다시 설정할 수 없습니다.

이름을 지정한 후에는 클라이언트를 다시 등록할 수 없습니다. 이는 ID 관리에서 LDAP의 클라이언트 항목의 키 속성이 클라이언트의 호스트 이름, FQDN 이기 때문입니다. 클라이언트의 LDAP 오브젝트를 다시 설정하는 것과는 달리 클라이언트의 이름 변경 결과는 클라이언트에 새 FQDN 을 사용하여 다른 LDAP 오브젝트에 키 및 기타 정보가 있다는 것입니다. 따라서 클라이언트 이름을 바꾸는 유일한 방법은 IdM에서 호스트를 제거하고 호스트의 호스트 이름을 변경한 후 IdM 클라이언트에 새 이름으로 설치하는 것입니다. 클라이언트의 이름을 바꾸는 방법에 대한 자세한 내용은 [Identity Management 클라이언트 시스템 복원](#)을 참조하십시오.

클라이언트 재등록 중 어떤 일이 발생합니까?

Identity Management를 다시 설정하는 동안 다음을 수행합니다.

- 원래 호스트 인증서 취소

- 새 SSH 키 생성
- 새 키맵 생성

### 30.9.2. 사용자 인증 정보를 사용하여 클라이언트 등록: 대화형 재등록

이 절차에서는 권한이 부여된 사용자의 자격 증명을 사용하여 대화식으로 ID 관리 클라이언트를 다시 등록하는 방법을 설명합니다.

1. 호스트 이름이 동일한 클라이언트 시스템을 다시 생성합니다.
2. 클라이언트 시스템에서 **ipa-client-install --force-join** 명령을 실행합니다.

```
# ipa-client-install --force-join
```

3. 이 스크립트는 ID를 사용하여 클라이언트를 다시 설정하는 데 사용할 사용자를 묻는 메시지를 표시합니다. 예를 들어 **Enrollment Administrator** 역할이 있는 **hostadmin** 사용자일 수 있습니다.

```
User authorized to enroll computers: hostadmin
Password for hostadmin@EXAMPLE.COM:
```

#### 추가 리소스

- ***Identity Management** 설치에서 사용자 **credentials: Interactive install**을 사용하여 클라이언트 설치를 참조하십시오.*

### 30.9.3. 클라이언트 키맵을 사용하여 클라이언트 인증: 비대화형 재등록

#### 사전 요구 사항

- **/tmp** 또는 **/root** 디렉토리 및 같이 원래 클라이언트 키맵 파일을 백업합니다.

#### 절차

이 절차에서는 클라이언트 시스템의 키맵을 사용하여 **IdM(Identity Management)** 클라이언트를 비대화형으로 다시 설정하는 방법을 설명합니다. 예를 들어 클라이언트 키맵을 사용하여 다시 등록하면 자동

화된 설치에 적합합니다.

1. 호스트 이름이 동일한 클라이언트 시스템을 다시 생성합니다.
2. 백업 위치의 키탭 파일을 다시 생성된 클라이언트 시스템의 **/etc/** 디렉터리에 복사합니다.
3. **ipa-client-install** 유틸리티를 사용하여 클라이언트를 다시 설정한 후 **--keytab** 옵션으로 **keytab** 위치를 지정합니다.

```
# ipa-client-install --keytab /etc/krb5.keytab
```



참고

**--keytab** 옵션에 지정된 **keytab**은 등록을 시작하기 위해 인증하는 경우에만 사용됩니다. 재등록 중에 **IdM**은 클라이언트에 대한 새 키탭을 생성합니다.

#### 30.9.4. 설치 후 Identity Management 클라이언트 테스트

명령줄 인터페이스를 사용하면 **ipa-client-install** 이 성공했지만 자체 테스트를 수행할 수도 있습니다.

**Identity Management** 클라이언트에서 서버에 정의된 사용자에 대한 정보를 얻을 수 있는지 테스트하려면 서버에 정의된 사용자를 확인할 수 있는지 확인합니다. 예를 들어 기본 **admin** 사용자를 확인하려면 다음을 실행합니다.

```
[user@client1 ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

인증이 올바르게 작동하는지 테스트하려면 **su -** 다른 **IdM** 사용자로 다음을 수행합니다.

```
[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 ~]$
```

#### 30.10. ID 관리 클라이언트 시스템 이름 변경

다음 섹션에서는 **ID** 관리 클라이언트 시스템의 호스트 이름을 변경하는 방법에 대해 설명합니다.



## 주의

클라이언트 이름 변경은 수동 절차입니다. 호스트 이름을 변경하지 않는 한 반드시 수행하지 마십시오.

ID 관리 클라이언트 이름을 변경하려면 다음이 포함됩니다.

1. 호스트 준비 자세한 내용은 [IdM 클라이언트 이름 변경 준비를 참조하십시오](#).
2. 호스트에서 **IdM** 클라이언트 설치 제거. 자세한 내용은 [ID 관리 클라이언트 설치 제거를 참조하십시오](#).
3. 호스트 이름 변경 자세한 내용은 [호스트 시스템 복원을 참조하십시오](#).
4. 새 이름으로 호스트에 **IdM** 클라이언트 설치. 자세한 내용은 [ID 관리 설치 시 ID 관리 클라이언트: 기본 시나리오 설치](#)를 참조하십시오.
5. **IdM** 클라이언트 설치 후 호스트 구성. 자세한 내용은 [서비스 다시 생성, 인증서 재 생성 및 호스트 그룹 다시 추가](#)를 참조하십시오.

### 30.10.1. IdM 클라이언트 이름 변경 준비

현재 클라이언트를 제거하기 전에 클라이언트에 대한 특정 설정을 기록합니다. 새 호스트 이름으로 시스템을 다시 로드한 후 이 구성을 적용합니다.

- 시스템에서 실행 중인 서비스를 확인합니다.
  - `ipa service-find` 명령을 사용하고 출력에서 인증서로 서비스를 식별합니다.

```
$ ipa service-find old-client-name.example.com
```

○

또한 각 호스트에는 **ipa service-find** 출력에 표시되지 않는 기본 호스트 서비스가 있습니다. 호스트 주체라고도 하는 호스트 서비스의 서비스 주체는 **host/old-client-name.example.com** 입니다.

●

**ipa service-find old-client-name.example.com** 에서 표시되는 모든 서비스 주체의 경우 **old-client-name.example.com** 시스템에서 해당 키탭의 위치를 확인합니다.

```
# find / -name "*.keytab"
```

클라이언트 시스템의 각 서비스에는 **service\_name/host\_name@REALM** 형식의 Kerberos 사용자가 있습니다(예: **ldap/old-client-name.example.com@EXAMPLE.COM**).

●

시스템이 속한 모든 호스트 그룹을 식별합니다.

```
# ipa hostgroup-find old-client-name.example.com
```

### 30.10.2. ID 관리 클라이언트 설치 제거

클라이언트 설치 제거는 **SSSD(System Security Services Daemon)**와 같은 시스템 서비스의 모든 특정 ID 관리 구성과 함께 ID 관리 도메인에서 클라이언트를 제거합니다. 이렇게 하면 클라이언트 시스템의 이전 구성이 복원됩니다.

#### 절차

1.

**ipa-client-install --uninstall** 명령을 실행합니다.

```
[root@client]# ipa-client-install --uninstall
```

2.

서버에서 클라이언트 호스트의 **DNS** 항목을 수동으로 제거합니다.

```
[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.

`/etc/krb5.keytab` 이외의 각각의 식별된 키탭에 대해 이전 주체를 제거합니다.

```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

4.

**IdM** 서버에서 호스트 항목을 제거합니다. 이렇게 하면 모든 서비스가 제거되고 해당 호스트에 대해 발급된 모든 인증서가 취소됩니다.

```
[root@server ~]# ipa host-del client.example.com
```

### 30.10.3. 호스트 시스템 이름 변경

필요에 따라 머신의 이름을 바꿉니다. 예를 들어 다음과 같습니다.

```
[root@client]# hostnamectl set-hostname new-client-name.example.com
```

이제 새 호스트 이름을 사용하여 **Identity Management** 클라이언트를 **Identity Management** 도메인에 다시 설치할 수 있습니다.

### 30.10.4. 서비스 추가, 인증서 재 생성 및 호스트 그룹 다시 추가

**IdM(Identity Management)** 서버에서 이름 변경을 위해 **IdM 클라이언트 준비**에서 확인된 모든 서비스에 대해 새 키탭을 추가합니다.

+

```
[root@server ~]# ipa service-add service_name/new-client-name
```

1.

이름 변경을 위해 **IdM 클라이언트 준비**에 인증서가 할당된 서비스의 인증서를 생성합니다. 당신은 할 수 있습니다:

- **IdM** 관리 툴 사용
- **certmonger** 유틸리티 사용

2.

이름 변경을 위해 **IdM 클라이언트 준비**에서 확인된 호스트 그룹에 클라이언트를 다시 추가합

니다.

### 30.11. 호스트 항목 비활성화 및 다시 활성화

이 섹션에서는 **IdM(Identity Management)**에서 호스트를 비활성화하고 다시 활성화하는 방법을 설명합니다.

#### 30.11.1. 호스트 비활성화

**IdM**에서 호스트 항목을 비활성화하려면 다음 절차를 완료합니다.

도메인 서비스, 호스트 및 사용자는 활성 호스트에 액세스할 수 있습니다. 예를 들어 유지 관리상의 이유로 활성 호스트를 일시적으로 제거해야 하는 경우가 있을 수 있습니다. 이러한 상황에서 호스트를 삭제하면 호스트 항목과 모든 관련 구성이 영구적으로 제거되므로 필요하지 않습니다. 대신 호스트 비활성화 옵션을 선택합니다.

호스트를 비활성화하면 도메인 사용자가 도메인에서 영구적으로 제거하지 않고 액세스할 수 없습니다.

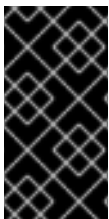
절차

•

**host-disable** 명령을 사용하여 호스트를 비활성화합니다. 호스트를 비활성화하면 호스트의 현재 활성 키맵이 종료됩니다. 예를 들어 다음과 같습니다.

```
$ kinit admin
$ ipa host-disable client.example.com
```

호스트를 비활성화하면 모든 **IdM** 사용자, 호스트 및 서비스에서 호스트를 사용할 수 없게 됩니다.



중요

호스트 항목을 비활성화하면 해당 호스트만 비활성화됩니다. 해당 호스트에서 구성된 모든 서비스도 비활성화됩니다.

#### 30.11.2. 호스트 다시 활성화

이 섹션에서는 비활성화된 **IdM** 호스트를 다시 활성화하는 방법에 대해 설명합니다.

호스트가 활성 **keytabs**를 비활성화하여 구성 항목을 변경하지 않고 **IdM** 도메인에서 호스트를 제거했습니다.

## 절차

- 호스트를 다시 활성화하려면 **ipa-getkeytab** 명령을 사용하여 다음을 추가합니다.
  - 키탭을 요청할 **IdM** 서버를 지정하는 **-s** 옵션
  - 보안 주체 이름을 지정하는 **-p** 옵션
  - **keytab**을 저장할 파일을 지정하는 **-k** 옵션입니다.

예를 들어 **client.example.com**에서 **server.example.com**에서 새 호스트 키탭을 요청하고 키탭을 **/etc/krb5.keytab** 파일에 저장하십시오.

```
$ ipa-getkeytab -s server.example.com -p host/client.example.com -k /etc/krb5.keytab -D
"cn=directory manager" -w password
```



## 참고

관리자의 자격 증명을 사용하여 **-D** "**uid=admin,cn=users,cn=accounts,dc=example,dc=com**" 을 지정할 수도 있습니다. 자격 증명은 호스트의 키탭을 만들 수 있는 사용자에게 해당하는 것이 중요합니다.

**ipa-getkeytab** 명령이 활성 **IdM** 클라이언트 또는 서버에서 실행된 경우, 를 사용하여 **TGT**(예: **kinit admin**)를 사용하여 가져온 **LDAP** 자격 증명(**-D** 및 **-w**) 없이 실행할 수 있습니다. 비활성화된 호스트에서 직접 명령을 실행하려면 **IdM** 서버에 인증할 **LDAP** 자격 증명을 제공합니다.

## 31장. IDM 웹 UI에서 호스트 항목 추가

이 장에서는 **IdM(Identity Management)**의 호스트와 **IdM 웹 UI**에 호스트 항목을 추가하는 작업을 소개합니다.

### 31.1. IDM의 호스트

**IdM(Identity Management)**은 이러한 **ID**를 관리합니다.

- 사용자
- 서비스
- 호스트

호스트는 시스템을 나타냅니다. **IdM ID**로, 호스트에 **IdM LDAP**에 대한 항목이 있으며, 이는 **IdM 서버**의 **389 Directory Server** 인스턴스입니다.

**IdM LDAP**의 호스트 항목은 다른 호스트와 도메인 내의 서비스 간 관계를 설정하는 데 사용됩니다. 이러한 관계는 도메인 내의 호스트에 대한 권한 부여 및 제어를 *위임*하는 과정의 일부입니다. 모든 호스트는 호스트 기반 액세스 제어 (**HBAC**) 규칙에 사용할 수 있습니다.

**IdM 도메인**은 일반 **ID** 정보, 공통 정책 및 공유 서비스를 통해 시스템 간의 공통성을 설정합니다. 도메인 클라이언트에 속하는 모든 시스템은 도메인의 클라이언트로 기능하므로 도메인에서 제공하는 서비스를 사용합니다. **IdM 도메인**은 특히 시스템용 세 가지 주요 서비스를 제공합니다.

- **DNS**
- **Kerberos**
- 인증서 관리

IdM의 호스트는 해당 호스트에서 실행되는 서비스와 밀접하게 연결됩니다.

- 서비스 항목은 호스트와 연결됩니다.
- 호스트는 호스트와 서비스 **Kerberos** 사용자를 모두 저장합니다.

## 31.2. 호스트 등록

이 섹션에서는 IdM 클라이언트로 호스트 등록 및 등록 후 수행되는 작업을 설명합니다. 섹션은 IdM 호스트 등록과 IdM 사용자를 비교합니다. 이 섹션에서는 호스트에서 사용할 수 있는 대체 인증 유형도 간략하게 설명합니다.

호스트 등록은 다음으로 구성됩니다.

- IdM LDAP에 호스트 항목 생성: IdM CLI에서 **ipa host-add** 명령 또는 이에 해당하는 IdM 웹 UI 작업을 사용할 수 있습니다.
- 호스트에서 IdM 서비스 구성(예: SSSD(System Security Services Daemon), Kerberos, certmonger 및 host를 IdM 도메인에 가입합니다.

두 작업은 개별적으로 또는 함께 수행할 수 있습니다.

별도로 수행하는 경우 서로 다른 수준의 권한을 가진 두 사용자 간에 두 작업을 분할할 수 있습니다. 이는 대규모 배포에 유용합니다.

**ipa-client-install** 명령은 두 가지 작업을 함께 수행할 수 있습니다. 이 명령은 해당 항목이 아직 없는 경우 IdM LDAP에 호스트 항목을 생성하고 호스트에 대한 **Kerberos** 및 **SSSD** 서비스 모두를 구성합니다. 이 명령은 IdM 도메인 내의 호스트를 가져와서 연결할 IdM 서버를 식별할 수 있습니다. 호스트가 IdM에서 관리하는 DNS 영역에 속하는 경우 **ipa-client-install** 은 호스트에 대한 DNS 레코드도 추가합니다. 명령은 클라이언트에서 실행해야 합니다.

## 31.3. 호스트 등록에 필요한 사용자 권한

호스트 등록 작업을 수행하려면 권한이 없는 사용자가 IdM 도메인에 원치 않는 시스템을 추가하지 못

하도록 인증을 수행해야 합니다. 필요한 권한은 다음과 같은 여러 요인에 따라 달라집니다.

- 호스트 항목이 **ipa-client-install** 실행과 별도로 생성되는 경우
- 등록에 일회성 암호(OTP)를 사용하는 경우

#### IdM LDAP에서 호스트 항목을 수동으로 생성하는 선택 사항의 사용자 권한

**ipa host-add CLI** 명령 또는 **IdM 웹 UI**를 사용하여 **IdM LDAP**에 호스트 항목을 생성하는 데 필요한 사용자 권한은 호스트 관리자입니다. 호스트 관리자 권한은 **IT professionals** 역할을 통해 얻을 수 있습니다.

#### 클라이언트를 IdM 도메인에 가입하기 위한 사용자 권한

호스트는 **ipa-client-install** 명령을 실행하는 동안 **IdM** 클라이언트로 구성됩니다. **ipa-client-install** 명령을 실행하는 데 필요한 인증 정보 수준은 다음 중 사용자가 직접 찾을 수 있는 시나리오에 따라 다릅니다.

- **IdM LDAP**의 호스트 항목이 없습니다. 이 시나리오에서는 전체 관리자의 자격 증명 또는 호스트 관리자 역할이 필요합니다. 전체 관리자는 **admins** 그룹의 멤버입니다. 호스트 관리자 역할은 호스트를 추가하고 호스트를 등록할 수 있는 권한을 제공합니다. 이 시나리오에 대한 자세한 내용은 [사용자 자격 증명을 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치.](#)
- **IdM LDAP**의 호스트 항목이 있습니다. 이 시나리오에서는 **ipa-client-install** 을 성공적으로 실행하려면 제한된 관리자의 인증 정보가 필요합니다. 이 경우 제한된 관리자에게는 **Enrollment Administrator** 역할이 있으며 이는 호스트 등록 권한을 제공합니다. 자세한 내용은 [사용자 자격 증명을 사용하여 클라이언트 설치: 대화식 설치.](#)
- **IdM LDAP**의 호스트 항목이 있으며 전체 또는 제한된 관리자가 호스트에 대한 **OTP**가 생성되었습니다. 이 시나리오에서는 **ipa-client-install** 명령을 **--password** 옵션과 함께 실행하여 올바른 **OTP**를 제공하는 경우 일반 사용자로 **IdM** 클라이언트를 설치할 수 있습니다. 자세한 내용은 [일회성 암호: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오.](#)

등록 후 **IdM** 호스트는 모든 새 세션을 인증하여 **IdM** 리소스에 액세스할 수 있습니다. **IdM** 서버가 시스템을 신뢰하고 해당 시스템에 설치된 클라이언트 소프트웨어에서 **IdM** 연결을 수락하려면 시스템 인증이 필요합니다. 클라이언트를 인증한 후 **IdM** 서버는 해당 요청에 응답할 수 있습니다.

### 31.4. IDM 호스트 및 사용자의 등록 및 인증: 비교

IdM의 사용자와 호스트 간에는 많은 유사성이 있습니다. 이 섹션에서는 등록 단계에서 확인할 수 있는 유사성과 배포 단계에서 인증과 관련된 몇 가지에 대해 설명합니다.

- 등록 단계(사용자 및 호스트 등록):
  - 관리자는 사용자 또는 호스트가 실제로 IdM에 참여하기 전에 사용자와 호스트에 대한 LDAP 항목을 생성할 수 있습니다. stage 사용자의 경우 명령은 `ipa stageuser-add` 입니다. host는 `ipa host-add` 입니다.
  - 사용자 암호를 어느 정도까지 줄일 수 있는 키 테이블 또는 축약, 키맵, 대칭 키가 포함된 파일은 호스트에서 `ipa-client-install` 명령을 실행하는 동안 생성됩니다. 이로 인해 호스트가 IdM 영역에 결합됩니다. Analogically는 사용자가 계정을 활성화할 때 암호를 생성해야 하므로 IdM 영역에 가입해야 합니다.
  - 사용자 암호는 사용자의 기본 인증 방법이지만 keytab은 호스트의 기본 인증 방법입니다. keytab은 호스트의 파일에 저장됩니다.

표 31.1. 사용자 및 호스트 등록

| 동작     | 사용자                                                     | 호스트                                                                  |
|--------|---------------------------------------------------------|----------------------------------------------------------------------|
| 사전 등록  | <code>\$ ipa stageuser-add user_name [-password]</code> | <code>\$ ipa host-add host_name [--random]</code>                    |
| 계정 활성화 | <code>\$ ipa stageuser-activate user_name</code>        | <code>\$ IPA-client install [--password]</code><br>(호스트 자체에서 실행해야 함) |

- 배포 단계(사용자 및 호스트 세션 인증)
  - 사용자가 새 세션을 시작하면 사용자는 암호를 사용하여 인증합니다. 마찬가지로, 호스트가 키맵 파일을 표시하여 인증합니다. SSSD(System Security Services Daemon)는 백그라운드에서 이 프로세스를 관리합니다.
  - 인증에 성공하면 사용자 또는 호스트는 TGT(Kerberos 티켓 부여 티켓)를 가져옵니다.
  - TGT는 특정 서비스에 대한 특정 티켓을 얻는 데 사용됩니다.

표 31.2. 사용자 및 호스트 세션 인증

| 사용자           |                                      | 호스트                                  |
|---------------|--------------------------------------|--------------------------------------|
| 기본 인증 수단      | 암호                                   | keytabs                              |
| 세션 시작(최신 사용자) | \$ kinit <i>user_name</i>            | [Winding on the host]                |
| 성공적인 인증의 결과   | TGT 는 특정 서비스에 대한 액세스 권한을 얻는 데 사용됩니다. | TGT 는 특정 서비스에 대한 액세스 권한을 얻는 데 사용됩니다. |

TGT 및 기타 Kerberos 티켓은 서버에서 정의한 Kerberos 서비스 및 정책의 일부로 생성됩니다. Kerberos 티켓의 초기 부여, Kerberos 자격 증명 갱신, Kerberos 세션 삭제도 IdM 서비스에서 자동으로 처리됩니다.

### IdM 호스트에 대한 대체 인증 옵션

IdM은 키탭 외에도 두 가지 유형의 시스템 인증을 지원합니다.

- SSH 키. 호스트의 SSH 공개 키가 생성되어 호스트 항목에 업로드됩니다. 여기에서 SSSD(System Security Services Daemon)는 IdM을 ID 공급자로 사용하며 OpenSSH 및 기타 서비스와 협력하여 IdM에 있는 공개 키를 참조할 수 있습니다.
- 시스템 인증서. 이 경우 시스템은 IdM 서버의 인증 기관에서 발행한 SSL 인증서를 사용한 다음 IdM 디렉터리 서버에 저장됩니다. 그런 다음 인증서를 서버에 인증할 때 존재하는 시스템으로 전송됩니다. 클라이언트에서 인증서는 [certmonger](#) 라는 서비스로 관리됩니다.

## 31.5. IDM LDAP의 호스트 항목

이 섹션에서는 IdM(Identity Management)의 호스트 항목과 포함할 수 있는 특성에 대해 설명합니다.

LDAP 호스트 항목에는 IdM 내의 클라이언트에 대한 모든 관련 정보가 포함되어 있습니다.

- 호스트와 관련된 서비스 항목
- 호스트 및 서비스 주체

- 액세스 제어 규칙
- 물리적 위치 및 운영 체제와 같은 머신 정보



## 참고

**IdM 웹 UI ID** → 호스트 탭에 **IdM LDAP**에 저장된 특정 호스트에 대한 모든 정보가 표시되지 않습니다.

## 호스트 항목 구성 속성

호스트 항목은 물리적 위치, **MAC** 주소, 키 및 인증서와 같이 시스템 구성 외부에 있는 호스트에 대한 정보를 포함할 수 있습니다.

이 정보는 수동으로 생성되는 경우 호스트 항목을 생성할 때 설정할 수 있습니다. 또는 이 정보의 대부분은 호스트가 도메인에 등록된 후 호스트 항목에 추가할 수 있습니다.

표 31.3. 호스트 구성 속성

| UI 필드    | 명령줄 옵션                       | 설명                                                                                 |
|----------|------------------------------|------------------------------------------------------------------------------------|
| 설명       | <b>--desc</b> =description   | 호스트에 대한 설명입니다.                                                                     |
| 지역성      | <b>--locality</b> =locality  | 호스트의 지리적 위치입니다.                                                                    |
| 위치       | <b>--location</b> =location  | 데이터 센터 랙과 같은 호스트의 물리적 위치.                                                          |
| 플랫폼      | <b>--platform</b> =string    | 호스트 하드웨어 또는 아키텍처.                                                                  |
| 운영 체제    | <b>--os</b> =string          | 호스트의 운영 체제 및 버전입니다.                                                                |
| MAC 주소   | <b>--macaddress</b> =address | 호스트의 MAC 주소입니다. 이는 다중 값 특성입니다. MAC 주소는 NIS 플러그인에서 호스트에 대한 NIS 이더넷 맵을 생성하는 데 사용됩니다. |
| SSH 공개 키 | <b>--sshpkey</b> =string     | 호스트의 전체 SSH 공개 키입니다. 이는 다중 값 특성이므로 여러 키를 설정할 수 있습니다.                               |

| UI 필드               | 명령줄 옵션                           | 설명                                                                                                                                               |
|---------------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 보안 주체 이름( 편집할 수 없음) | <b>--principalname=principal</b> | 호스트에 대한 Kerberos 사용자 이름입니다. 다른 보안 주체가 <b>-p</b> 에 명시적으로 설정되어 있지 않은 경우 클라이언트 설치 중에 기본적으로 호스트 이름으로 설정됩니다. 명령줄 도구를 사용하여 변경할 수 있지만 UI에서는 변경할 수 없습니다. |
| 일회성 암호 설정           | <b>--password=string</b>         | 이 옵션은 대규모 등록에 사용할 수 있는 호스트의 암호를 설정합니다.                                                                                                           |
| -                   | <b>--random</b>                  | 이 옵션은 대규모 등록에 사용할 임의의 암호를 생성합니다.                                                                                                                 |
| -                   | <b>--certificate=string</b>      | 호스트에 대한 인증서 Blob입니다.                                                                                                                             |
| -                   | <b>--updatedns</b>               | 이렇게 하면 IP 주소가 변경되면 호스트가 DNS 항목을 동적으로 업데이트할 수 있는지 여부가 설정됩니다.                                                                                      |

### 31.6. 웹 UI에서 호스트 항목 추가

1. **Identity (ID)** 탭을 열고 **Hosts (호스트)** 하위 탭을 선택합니다.
2. 호스트 목록 상단에서 **Add (추가)**를 클릭합니다.

그림 31.1. 호스트 항목 추가

Hosts

Search

Refresh Delete **+Add** Actions ▾

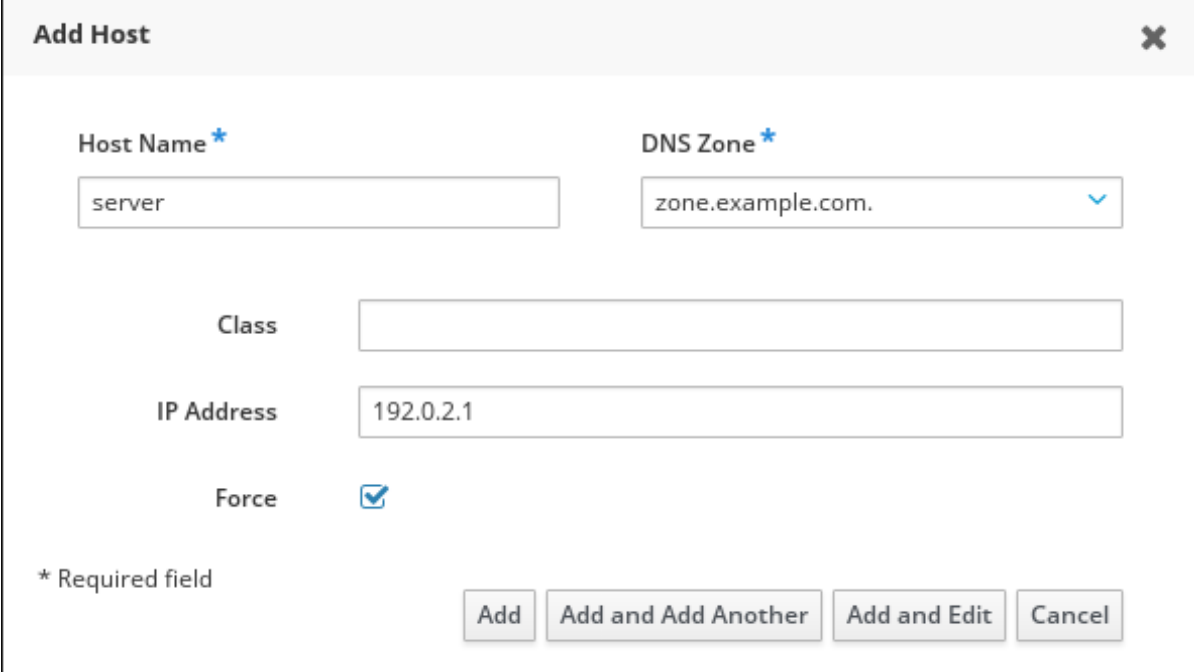
| <input type="checkbox"/> | Host name          | Description | Enrolled |
|--------------------------|--------------------|-------------|----------|
| <input type="checkbox"/> | server.example.com |             | True     |

Showing 1 to 1 of 1 entries.

3. 시스템 이름을 입력하고 구성된 영역에서 도메인을 드롭다운 목록에 선택합니다. 호스트에 이미 고정 IP 주소가 할당된 경우 **DNS** 항목이 완전히 생성되도록 **host** 항목과 함께 호스트를 포함합니다.

현재 클래스 필드는 특정 목적을 가지고 있지 않습니다.

그림 31.2. 호스트 마법사 추가



The 'Add Host' dialog box contains the following fields and controls:

- Host Name \***: Text input field containing 'server'.
- DNS Zone \***: Dropdown menu showing 'zone.example.com.' with a blue checkmark.
- Class**: Empty text input field.
- IP Address**: Text input field containing '192.0.2.1'.
- Force**: Checkmark button (checked).
- \* Required field**: Text label.
- Buttons**: 'Add', 'Add and Add Another', 'Add and Edit', and 'Cancel'.

IdM에 **DNS** 영역을 생성할 수 있습니다. IdM 서버가 **DNS** 서버를 관리하지 않는 경우 일반 텍스트 필드와 같이 메뉴 영역에 영역을 수동으로 입력할 수 있습니다.



#### 참고

호스트가 **DNS**를 통해 확인할 수 있는지 여부를 확인하려면 **Force** 확인란을 선택합니다.

4.

추가 및 편집 버튼을 클릭하여 확장된 항목 페이지로 직접 이동하여 추가 특성 정보를 입력합니다. 호스트 하드웨어 및 물리적 위치에 대한 정보를 호스트 항목에 포함할 수 있습니다.

그림 31.3. 확장 엔트리 페이지

Host: server.zone.example.com

server.zone.examp... is a member of:

Settings

Host Groups

Netgroups

Roles

HBAC Rules

Sudo Rules

ser

Refresh

Revert

Save

Actions

Host Settings

Host name

server.zone.example.com

Principal name

host/server.zone.example.com@EXAMPLE.COM

Description

Class

Locality

## 32장. ANSIBLE 플레이북을 사용하여 호스트 관리

**Ansible**은 시스템 구성, 소프트웨어 배포 및 롤링 업데이트를 수행하는 데 사용되는 자동화 틀입니다. **Ansible**에는 **IdM(Identity Management)** 지원이 포함되어 있으며 **Ansible** 모듈을 사용하여 호스트 관리를 자동화할 수 있습니다.

이 장에서는 **Ansible** 플레이북을 사용하여 호스트 및 호스트 항목을 관리할 때 수행되는 다음 개념과 작업을 설명합니다.

- **FQDN**으로만 정의된 **IdM** 호스트 항목이 있는지 확인
- **IP** 주소를 사용하여 **IdM** 호스트 항목이 있는지 확인
- 임의의 암호로 여러 **IdM** 호스트 항목이 있는지 확인
- 여러 **IP** 주소로 **IdM** 호스트 항목이 있는지 확인
- **IdM** 호스트 항목이 없는지 확인

### 32.1. ANSIBLE 플레이북을 사용하여 FQDN으로 IDENTITY 호스트 항목이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 항목이 있는지 확인하는 방법을 설명합니다. 호스트 항목은 **FQDN**( 정규화된 도메인 이름 )으로만 정의됩니다.

다음 조건 중 하나가 적용되는 경우 호스트의 **FQDN** 이름을 지정하면 충분합니다.

- **IdM** 서버는 **DNS**를 관리하도록 구성되지 않았습니다.
- 호스트에 고정 **IP** 주소가 없거나 호스트가 구성된 시점에 **IP** 주소를 알 수 없습니다. **FQDN** 으로만 정의된 호스트를 추가하면 기본적으로 **IdM DNS** 서비스에 자리 표시자 항목이 생성됩니다. 예를 들어 랩탑은 **IdM** 클라이언트로 사전 구성될 수 있지만, 구성된 시점에는 **IP** 주소가 없습니다. **DNS** 서비스가 레코드를 동적으로 업데이트하면 호스트의 현재 **IP** 주소가 감지되고 해당 **DNS** 레코드가 업데이트됩니다.



## 참고

Ansible이 없으면 `ipa host-add` 명령을 사용하여 IdM에 호스트 항목이 생성됩니다. IdM에 호스트를 추가하는 결과는 IdM에 호스트 상태가 됩니다. Ansible의 **idempotence on idempotence** 때문에 Ansible을 사용하여 호스트를 IdM에 추가하려면 호스트 상태를 **present: state: present** 로 정의하는 플레이북을 생성해야 합니다.

## 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- **ansible-freeipa** 패키지는 Ansible 컨트롤러에 설치됩니다.

## 절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 `ipaserver` 를 이 파일에 정의합니다.

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

2. 확인하려는 IdM에 있는 호스트의 FQDN 으로 Ansible 플레이북 파일을 생성합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/host/add-host.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Host present
  hosts: ipaserver
  become: true

  tasks:
    - name: Host host01.idm.example.com present
      ipahost:
        ipadmin_password: MySecret123
        name: host01.idm.example.com
        state: present
        force: yes
```

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-host-is-present.yml
```



## 참고

이 절차를 수행하면 **IdM LDAP** 서버의 호스트 항목이 생성되지만 **IdM Kerberos** 영역에 호스트를 등록하지 않습니다. 이를 위해 호스트를 **IdM** 클라이언트로 배포해야 합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 ID 관리 클라이언트 설치](#)를 참조하십시오.

## 검증 단계

1.

**IdM** 서버에 **admin**으로 로그인합니다.

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

2.

**ipa host-show** 명령을 입력하고 호스트 이름을 지정합니다.

```
$ 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
```

출력은 **IdM**에 **host01.idm.example.com** 이 있는지 확인합니다.

## 32.2. ANSIBLE 플레이북을 사용하여 DNS 정보로 IDM 호스트 항목이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 항목이 있는지 확인하는 방법을 설명합니다. 호스트 항목은 **FQDN**( 정규화된 도메인 이름 ) 및 해당 **IP** 주소로 정의됩니다.



## 참고

**Ansible**이 없으면 **ipa host-add** 명령을 사용하여 **IdM**에 호스트 항목이 생성됩니다. **IdM**에 호스트를 추가하는 결과는 **IdM**에 호스트 상태가 됩니다. **Ansible**의 **idempotence on idempotence** 때문에 **Ansible**을 사용하여 호스트를 **IdM**에 추가하려면 호스트 상태를 **present: state: present** 로 정의하는 플레이북을 생성해야 합니다.

## 사전 요구 사항

•

**IdM** 관리자 암호를 알고 있습니다.

•

**ansible-freeipa** 패키지는 **Ansible** 컨트롤러에 설치됩니다.

## 절차

1.

인벤토리 파일(예: **inventory.file**)을 생성하고 **ipaserver** 를 이 파일에 정의합니다.

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

2.

보장하려는 **IdM**에 있는 호스트의 **FQDN**( 정규화된 도메인 이름 )을 사용하여 **Ansible** 플레이북 파일을 생성합니다. 또한 **IdM** 서버가 **DNS**를 관리하도록 구성되어 호스트의 **IP** 주소를 알고 있는 경우 **ip\_address** 매개 변수의 값을 지정합니다. 호스트가 **DNS** 리소스 레코드에 존재하려면 **IP** 주소가 필요합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/host/host-present.yml** 파일에서 예제를 복사하고 수정할 수 있습니다. 다른 추가 정보를 포함할 수도 있습니다.

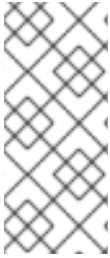
```
---
- name: Host present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure host01.idm.example.com is present
    ipahost:
      ipaadmin_password: MySecret123
      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.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-host-is-present.yml
```



## 참고

이 절차를 수행하면 **IdM LDAP** 서버의 호스트 항목이 생성되지만 **IdM Kerberos** 영역에 호스트를 등록하지 않습니다. 이를 위해 호스트를 **IdM** 클라이언트로 배포해야 합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 ID 관리 클라이언트 설치](#)를 참조하십시오.

## 검증 단계

1.

**IdM** 서버에 **admin**으로 로그인합니다.

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

2.

**ipa host-show** 명령을 입력하고 호스트 이름을 지정합니다.

```
$ 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
```

출력은 **IdM**에 **host01.idm.example.com** 이 있는지 확인합니다.

### 32.3. ANSIBLE 플레이북을 사용하여 임의의 암호로 여러 IDМ 호스트 항목이 있는지 확인

**ipahost** 모듈을 사용하면 시스템 관리자가 하나의 **Ansible** 작업만 사용하여 **IdM**에 여러 호스트 항목이 있거나 있는지 확인할 수 있습니다. 이 섹션에서는 **FQDN**( 정규화된 도메인 이름 )에서만 정의한 여러 호스트 항목이 있는지 확인하는 방법을 설명합니다. **Ansible** 플레이북을 실행하면 호스트에 대한 임의의 암호가 생성됩니다.



## 참고

Ansible이 없으면 `ipa host-add` 명령을 사용하여 IdM에 호스트 항목이 생성됩니다. IdM에 호스트를 추가하는 결과는 IdM에 호스트 상태가 됩니다. Ansible의 **idempotence on idempotence** 때문에 Ansible을 사용하여 호스트를 IdM에 추가하려면 호스트 상태를 **present: state: present** 로 정의하는 플레이북을 생성해야 합니다.

## 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- **ansible-freeipa** 패키지는 Ansible 컨트롤러에 설치됩니다.

## 절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 `ipaserver` 를 이 파일에 정의합니다.

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

2. 보장하려는 IdM에 있는 호스트의 **FQDN**( 정규화된 도메인 이름 )을 사용하여 Ansible 플레이북 파일을 생성합니다. Ansible 플레이북에서 호스트가 IdM에 이미 있고 `update_password` 가 `on_create` 로 제한되더라도 각 호스트에 대해 임의의 암호를 생성하도록 하려면 `random: yes` 및 `force: yes` 옵션을 추가합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/README-host.md` Markdown 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Ensure hosts with random password
  hosts: ipaserver
  become: true

  tasks:
    - name: Hosts host01.idm.example.com and host02.idm.example.com present with
      random passwords
      ipahost:
        ipaadmin_password: MySecret123
        hosts:
          - name: host01.idm.example.com
            random: yes
            force: yes
          - name: host02.idm.example.com
            random: yes
            force: yes
      register: ipahost
```

3.

플레이북을 실행합니다.

```
$ ansible-playbook -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": "0HolRvjUdH0Ycbf6uYdWTxH"},
"host02.idm.example.com": {"randompassword": "5VdLgrf3wvojmACdHC3uA3s"}}
```



참고

임의 일회성 암호(OTP)를 사용하여 호스트를 IdM 클라이언트로 배포하려면 [Ansible 플레이북을 사용하여 IdM 클라이언트 등록에 대한 권한 부여 옵션](#)을 참조하거나 일회성 암호: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오.

### 검증 단계

1.

IdM 서버에 admin으로 로그인합니다.

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

2.

ipa host-show 명령을 입력하고 호스트 중 하나의 이름을 지정합니다.

```
$ ipa host-show host01.idm.example.com
Host name: host01.idm.example.com
Password: True
Keytab: False
Managed by: host01.idm.example.com
```

출력에서는 임의의 암호를 사용하여 IdM에 host01.idm.example.com 이 있는지 확인합니다.

## 32.4. ANSIBLE 플레이북을 사용하여 여러 IP 주소로 IDM 호스트 항목이 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 호스트 항목이 있는지 확인하는 방법을 설명합니다. 호스트 항목은 FQDN( 정규화된 도메인 이름 )과 해당 다중 IP 주소로 정의됩니다.



## 참고

**ipa host** 유틸리티와 달리 **Ansible ipahost** 모듈은 호스트에 여러 **IPv4** 및 **IPv6** 주소가 있는지 확인할 수 있습니다. **ipa host-mod** 명령은 **IP** 주소를 처리할 수 없습니다.

## 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **ansible-freeipa** 패키지는 **Ansible** 컨트롤러에 설치됩니다.

## 절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 **ipaserver** 를 이 파일에 정의합니다.

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

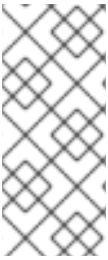
2. **Ansible** 플레이북 파일을 생성합니다. **ipahost** 변수의 이름으로, 보장하려는 **IdM**에 있는 호스트의 **FQDN**(정규화된 도메인 이름)을 지정합니다. **ip\_address** 구문을 사용하여 별도의 행에 여러 **IPv4** 및 **IPv6** **ip\_address** 값을 지정합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/host/host-member-ipaddresses-present.yml** 파일에서 예제를 복사하고 수정할 수 있습니다. 추가 정보를 포함할 수도 있습니다.

```
---
- name: Host member IP addresses present
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure host101.example.com IP addresses present
      ipahost:
        ipadmin_password: MySecret123
        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. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-host-with-multiple-IP-addresses-is-present.yml
```



## 참고

이 절차에서는 **IdM LDAP** 서버에 호스트 항목을 생성하지만, **IdM Kerberos** 영역에 호스트를 등록하지 않습니다. 이를 위해 호스트를 **IdM** 클라이언트로 배포해야 합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 ID 관리 클라이언트 설치](#)를 참조하십시오.

## 검증 단계

1.

**IdM** 서버에 **admin**으로 로그인합니다.

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

2.

**ipa host-show** 명령을 입력하고 호스트 이름을 지정합니다.

```
$ 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
```

출력은 **IdM**에 **host01.idm.example.com** 이 있는지 확인합니다.

3.

**IdM DNS** 레코드에 호스트의 여러 **IP** 주소가 있는지 확인하려면 **ipa dnsHistory-show** 명령을 입력하고 다음 정보를 지정합니다.

- 

**IdM** 도메인의 이름

- 

호스트 이름

```
$ 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
```

출력에서 플레이북에 지정된 모든 IPv4 및 IPv6 주소가 `host01.idm.example.com` 호스트 항목과 올바르게 연결되었는지 확인합니다.

### 32.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 항목이 없는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에서 호스트 항목이 없는지 확인하는 방법을 설명합니다.

#### 사전 요구 사항

- IdM 관리자 인증 정보

#### 절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 `ipaserver` 를 이 파일에 정의합니다.

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

2. 보장하려는 IdM에 없는 호스트의 FQDN( 정규화된 도메인 이름 )을 사용하여 Ansible 플레이북 파일을 생성합니다. IdM 도메인에 DNS가 통합된 경우 `updatedns: yes` 옵션을 사용하여 DNS에서 호스트에 대한 모든 종류의 연결된 레코드를 제거합니다.

이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/host/delete-host.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Host absent
  hosts: ipaserver
  become: true

  tasks:
    - name: Host host01.idm.example.com absent
      ipahost:
        ipadmin_password: MySecret123
        name: host01.idm.example.com
        updatedns: yes
        state: absent
```

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-host-absent.yml
```

#### 참고

절차는 다음과 같습니다.

- **IdM Kerberos** 영역에는 호스트가 존재하지 않습니다.
- **IdM LDAP** 서버에 호스트 항목이 없습니다.

클라이언트 호스트 자체에서 **SSSD(System Security Services Daemon)**와 같은 시스템 서비스의 특정 **IdM** 구성을 제거하려면 클라이언트에서 **ipa-client-install --uninstall** 명령을 실행해야 합니다. 자세한 내용은 [IdM 클라이언트 설치 제거](#)를 참조하십시오.

#### 검증 단계

1. **admin**으로 **ipaserver**에 로그인합니다.

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

2. **host01.idm.example.com**에 대한 정보를 표시합니다.

```
$ ipa host-show host01.idm.example.com
ipa: ERROR: host01.idm.example.com: host not found
```

출력은 호스트가 **IdM**에 없음을 확인합니다.

#### 32.6. 추가 리소스

- **/usr/share/doc/ansible-freeipa/README-host.md** Markdown 파일을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/playbooks/host** 디렉터리에서 추가 플레이북을 참조하십시오.

### 33장. IDM CLI를 사용하여 호스트 그룹 관리

이 장에서는 **IdM(Identity Management)**의 호스트 그룹을 소개하고 **CLI(명령줄 인터페이스)**에서 호스트 그룹 및 해당 멤버를 관리하기 위해 다음 작업을 설명합니다.

- 호스트 그룹 및 해당 멤버 보기
- 호스트 그룹 생성
- 호스트 그룹 삭제
- 호스트 그룹 멤버 추가
- 호스트 그룹 멤버 제거
- 호스트 그룹 멤버 관리자 추가
- 호스트 그룹 멤버 관리자 제거

#### 33.1. IDM의 호스트 그룹

**IdM** 호스트 그룹은 중요한 관리 작업, 특히 액세스 제어에 대한 제어를 중앙 집중화하는 데 사용할 수 있습니다.

##### 호스트 그룹의 정의

호스트 그룹은 공통 액세스 제어 규칙 및 기타 특성을 가진 **IdM** 호스트 세트가 포함된 엔티티입니다. 예를 들어 회사 부서, 물리적 위치 또는 액세스 제어 요구 사항에 따라 호스트 그룹을 정의할 수 있습니다.

**IdM**의 호스트 그룹에는 다음이 포함될 수 있습니다.

- **IdM** 서버 및 클라이언트

- 기타 **IdM** 호스트 그룹

기본적으로 생성된 호스트 그룹

기본적으로 **IdM** 서버는 모든 **IdM** 서버 호스트에 대한 호스트 그룹 **ipaservers** 를 생성합니다.

직접 및 간접 그룹 멤버

**IdM**의 그룹 속성은 직접 및 간접 멤버 모두에 적용됩니다. 호스트 그룹 **B**가 호스트 그룹 **A**의 멤버인 경우 호스트 그룹 **B**의 모든 멤버가 호스트 그룹 **A**의 간접 멤버로 간주됩니다.

### 33.2. CLI를 사용하여 **IDM** 호스트 그룹 보기

이 섹션에서는 **CLI**(명령줄 인터페이스)를 사용하여 **IdM** 호스트 그룹을 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조 하십시오.](#)

절차

1. **ipa hostgroup-find** 명령을 사용하여 모든 호스트 그룹을 찾습니다.

```
$ ipa hostgroup-find
-----
1 hostgroup matched
-----
Host-group: ipaservers
Description: IPA server hosts
-----
Number of entries returned 1
-----
```

호스트 그룹의 모든 속성을 표시하려면 **--all** 옵션을 추가합니다. 예를 들어 다음과 같습니다.

```
$ 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-xxxxxxxxxxxxxx
objectclass: top, groupOfNames, nestedGroup, ipaobject, ipahostgroup
-----
```

```
Number of entries returned 1
-----
```

### 33.3. CLI를 사용하여 IDM 호스트 그룹 생성

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 **IdM** 호스트 그룹을 생성하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조](#) 하십시오.

#### 절차

1. **ipa hostgroup-add** 명령을 사용하여 호스트 그룹을 추가합니다.  
예를 들어 **group\_name** 이라는 **IdM** 호스트 그룹을 생성하고 설명을 제공하려면 다음을 수행합니다.

```
$ ipa hostgroup-add --desc 'My new host group' group_name
-----
Added hostgroup "group_name"
-----
Host-group: group_name
Description: My new host group
-----
```

### 33.4. CLI를 사용하여 IDM 호스트 그룹 삭제

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 **IdM** 호스트 그룹을 삭제하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **활성 Kerberos 티켓.** 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조 하십시오.](#)

## 절차

1.

**ipa hostgroup-del** 명령을 사용하여 호스트 그룹을 삭제합니다.  
예를 들어 **group\_name** 이라는 IdM 호스트 그룹을 삭제하려면 다음을 수행합니다.

```
$ ipa hostgroup-del group_name
-----
Deleted hostgroup "group_name"
-----
```



## 참고

그룹을 제거해도 IdM에서 그룹 멤버가 삭제되지 않습니다.

## 33.5. CLI를 사용하여 IDМ 호스트 그룹 멤버 추가

단일 명령을 사용하여 호스트 및 호스트 그룹을 IdM 호스트 그룹에 추가할 수 있습니다.

## 사전 요구 사항

- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **활성 Kerberos 티켓.** 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조 하십시오.](#)
- **선택사항입니다.** **ipa hostgroup-find** 명령을 사용하여 호스트 및 호스트 그룹을 찾습니다.

## 절차

1.

호스트 그룹에 멤버를 추가하려면 **ipa hostgroup-add-member** 를 사용하여 관련 정보를 제공합니다. 다음 옵션을 사용하여 추가할 멤버 유형을 지정할 수 있습니다.

•

IdM 호스트 그룹에 하나 이상의 호스트를 추가하려면 **--hosts** 옵션을 사용합니다. 예를 들어 **example\_member** 라는 호스트를 **group\_name** 이라는 그룹에 추가하려면 다음을 수행합니다.

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

•

호스트 그룹 옵션을 사용하여 IdM 호스트 그룹에 하나 이상의 호스트 그룹을 추가합니다. 예를 들어 **nested\_group** 이라는 호스트 그룹을 **group\_name** 이라는 그룹에 추가하려면 다음을 수행합니다.

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

•

다음 구문을 사용하여 하나의 단일 명령에서 여러 호스트와 여러 개의 호스트 그룹을 IdM 호스트 그룹에 추가할 수 있습니다.

```
$ ipa hostgroup-add-member group_name --hosts={host1,host2} --hostgroups={group1,group2}
```

#### 중요

다른 호스트 그룹의 멤버로 호스트 그룹을 추가할 때 재귀 그룹을 만들지 마십시오. 예를 들어 **Group A**가 **Group B**의 멤버인 경우 **그룹 B**를 **그룹 A**의 멤버로 추가하지 마십시오. 반복 그룹은 예기치 않은 동작이 발생할 수 있습니다.

### 33.6. CLI를 사용하여 IDM 호스트 그룹 멤버 제거

단일 명령을 사용하여 IdM 호스트 그룹에서 호스트 및 호스트 그룹을 제거할 수 있습니다.

## 사전 요구 사항

- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **활성 Kerberos 티켓.** 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조](#) 하십시오.
- **선택사항입니다.** `ipa hostgroup-find` 명령을 사용하여 그룹에 제거하려는 멤버가 포함되어 있는지 확인합니다.

## 절차

1. 호스트 그룹 멤버를 제거하려면 `ipa hostgroup-remove-member` 명령을 사용하고 관련 정보를 제공합니다. 다음 옵션을 사용하여 제거할 멤버 유형을 지정할 수 있습니다.

- **IdM 호스트 그룹에서 하나 이상의 호스트를 제거하려면 `--hosts` 옵션을 사용합니다.**  
예를 들어 `group_name` 이라는 그룹에서 `example_member` 라는 호스트를 제거하려면 다음을 수행합니다.

```
$ ipa hostgroup-remove-member group_name --hosts example_member
Host-group: group_name
Description: My host group
-----
Number of members removed 1
-----
```

- **IdM 호스트 그룹에서 하나 이상의 호스트 그룹을 제거하려면 `--hostgroups` 옵션을 사용합니다.**  
예를 들어 `group_name` 이라는 그룹에서 `nested_group` 이라는 호스트 그룹을 제거하려면 다음을 수행합니다.

```
$ ipa hostgroup-remove-member group_name --hostgroups example_member
Host-group: group_name
Description: My host group
-----
Number of members removed 1
-----
```



## 참고

그룹을 제거해도 **IdM**에서 그룹 멤버가 삭제되지 않습니다.

- 다음 구문을 사용하여 **IdM** 호스트 그룹에서 여러 호스트와 여러 호스트 그룹을 하나의 단일 명령으로 제거할 수 있습니다.

```
$ ipa hostgroup-remove-member group_name --hosts={host1,host2} --hostgroups={group1,group2}
```

### 33.7. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가

단일 명령을 사용하여 호스트 및 호스트 그룹을 멤버 관리자로 추가할 수 있습니다. 멤버 관리자는 **IdM** 호스트 그룹에 호스트 또는 호스트 그룹을 추가할 수 있지만 호스트 그룹의 속성은 변경할 수 없습니다.

#### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을 참조](#) 하십시오.
- 멤버 관리자로 추가하는 호스트 또는 호스트 그룹의 이름과 관리할 호스트 그룹의 이름이 있어야 합니다.

#### 절차

1. 선택사항입니다. **ipa hostgroup-find** 명령을 사용하여 호스트 및 호스트 그룹을 찾습니다.
2. 호스트 그룹에 멤버 관리자를 추가하려면 **ipa hostgroup-add-member-manager** 를 사용합니다.

예를 들어 **example\_member** 라는 사용자를 **member manager**로 **group\_name** 이라는 그룹에 추가하려면 다음을 수행합니다.

```
$ 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.

그룹 옵션을 사용하여 하나 이상의 호스트 그룹을 **IdM** 호스트 그룹에 구성원 관리자로 추가합니다.

예를 들어 **admin\_group** 이라는 호스트 그룹을 구성원 관리자로 **group\_name** 그룹에 추가하려면 다음을 수행합니다.

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



#### 참고

호스트 그룹에 멤버 관리자를 추가한 후 업데이트는 **ID** 관리 환경의 모든 클라이언트에 전파하는 데 시간이 걸릴 수 있습니다.

#### 검증 단계

•

**ipa group-show** 명령을 사용하여 호스트 사용자 및 호스트 그룹이 멤버 관리자로 추가되었는지 확인합니다.

```
$ 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
```

#### 추가 리소스

- 자세한 내용은 **ipa hostgroup-add-member-manager --help** 를 참조하십시오.
- 자세한 내용은 **ipa hostgroup-show --help** 를 참조하십시오.

### 33.8. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거

단일 명령을 사용하여 **IdM** 호스트 그룹에서 호스트 및 호스트 그룹을 **IdM** 호스트 그룹에서 제거할 수 있습니다. 멤버 관리자는 **IdM** 호스트 그룹에서 호스트 그룹 멤버 관리자를 제거할 수 있지만 호스트 그룹의 속성은 변경할 수 없습니다.

#### 사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [kinit를 사용하여 IdM에 수동으로 로그인하는 것을](#) 참조하십시오.
- 제거 중인 기존 멤버 관리자 호스트 그룹의 이름과 관리 중인 호스트 그룹의 이름이 있어야 합니다.

#### 절차

1. 선택사항입니다. **ipa hostgroup-find** 명령을 사용하여 호스트 및 호스트 그룹을 찾습니다.
2. 호스트 그룹에서 멤버 관리자를 제거하려면 **ipa hostgroup-remove-member-manager** 명령을 사용합니다.

예를 들어 **group\_name** 그룹의 멤버 관리자로 **example\_member** 라는 사용자를 제거하려면 다음을 수행합니다.

```
$ 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.

IdM 호스트 그룹에서 멤버 관리자로 하나 이상의 호스트 그룹을 제거하려면 **--groups** 옵션을 사용합니다.

예를 들어 *nested\_group* 이라는 호스트 그룹을 *group\_name* 그룹의 멤버 관리자로 제거하려면 다음을 수행합니다.

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



#### 참고

호스트 그룹에서 멤버 관리자를 제거한 후 업데이트는 ID 관리 환경의 모든 클라이언트에 전파되는 데 시간이 걸릴 수 있습니다.

#### 검증 단계

- **ipa group-show** 명령을 사용하여 호스트 사용자 및 호스트 그룹이 멤버 관리자로 제거되었는지 확인합니다.

```
$ ipa hostgroup-show group_name
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
```

#### 추가 리소스

- 자세한 내용은 **ipa hostgroup-remove-member-manager --help** 를 참조하십시오.
- 자세한 내용은 **ipa hostgroup-show --help** 를 참조하십시오.

## 34장. IDM 웹 UI를 사용하여 호스트 그룹 관리

이 장에서는 **IdM(Identity Management)**의 호스트 그룹을 소개하고 웹 인터페이스(**웹 UI**)에서 호스트 그룹 및 해당 멤버를 관리하기 위해 다음 작업을 설명합니다.

- 호스트 그룹 및 해당 멤버 보기
- 호스트 그룹 생성
- 호스트 그룹 삭제
- 호스트 그룹 멤버 추가
- 호스트 그룹 멤버 제거
- 호스트 그룹 멤버 관리자 추가
- 호스트 그룹 멤버 관리자 제거

### 34.1. IDM의 호스트 그룹

**IdM** 호스트 그룹은 중요한 관리 작업, 특히 액세스 제어에 대한 제어를 중앙 집중화하는 데 사용할 수 있습니다.

#### 호스트 그룹의 정의

호스트 그룹은 공통 액세스 제어 규칙 및 기타 특성을 가진 **IdM** 호스트 세트가 포함된 엔티티입니다. 예를 들어 회사 부서, 물리적 위치 또는 액세스 제어 요구 사항에 따라 호스트 그룹을 정의할 수 있습니다.

**IdM**의 호스트 그룹에는 다음이 포함될 수 있습니다.

- **IdM** 서버 및 클라이언트

- 기타 **IdM** 호스트 그룹

기본적으로 생성된 호스트 그룹

기본적으로 **IdM** 서버는 모든 **IdM** 서버 호스트에 대한 호스트 그룹 **ipaservers** 를 생성합니다.

직접 및 간접 그룹 멤버

**IdM**의 그룹 속성은 직접 및 간접 멤버 모두에 적용됩니다. 호스트 그룹 **B**가 호스트 그룹 **A**의 멤버인 경우 호스트 그룹 **B**의 모든 멤버가 호스트 그룹 **A**의 간접 멤버로 간주됩니다.

## 34.2. IDM 웹 UI에서 호스트 그룹 보기

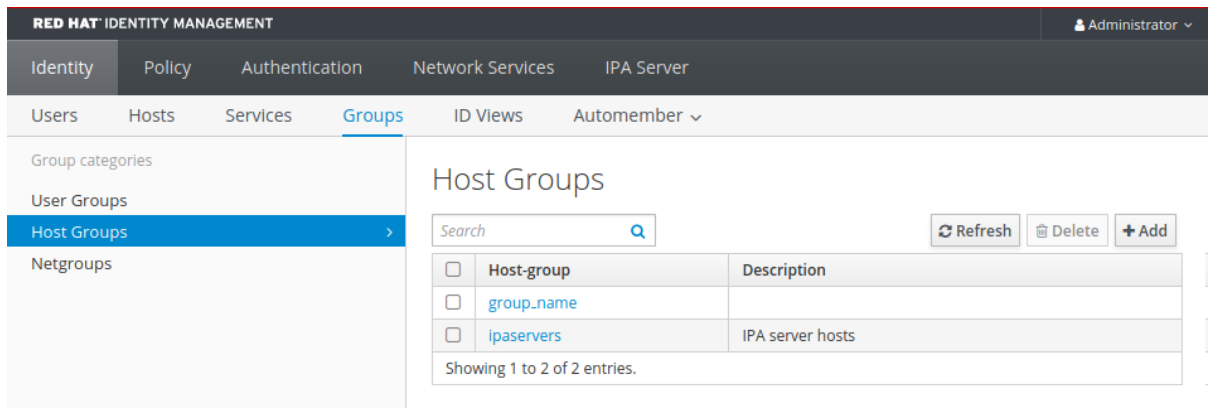
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 **IdM** 호스트 그룹을 보는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **IdM** 웹 UI에 로그인되어 있습니다. 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.

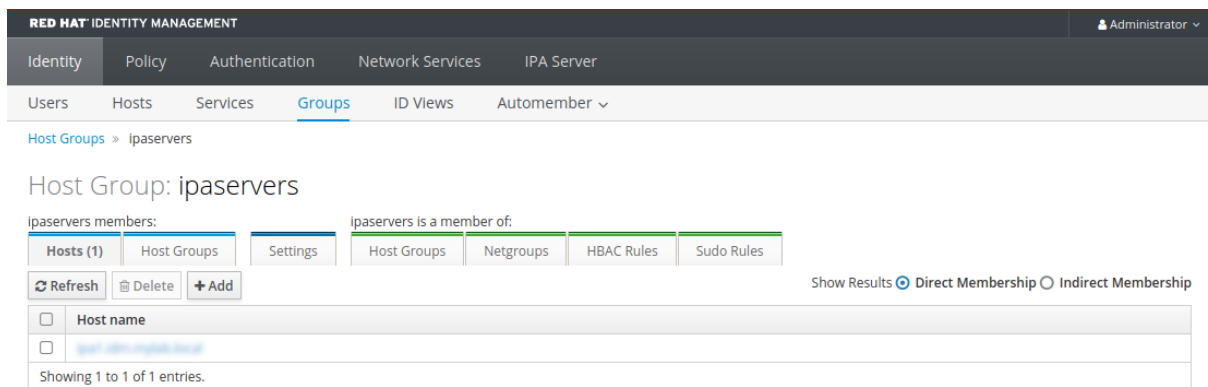
절차

1. **ID** → **그룹** 을 클릭하고 호스트 그룹 탭을 선택합니다.
  - 페이지에는 기존 호스트 그룹과 해당 설명이 나열됩니다.
  - 특정 호스트 그룹을 검색할 수 있습니다.



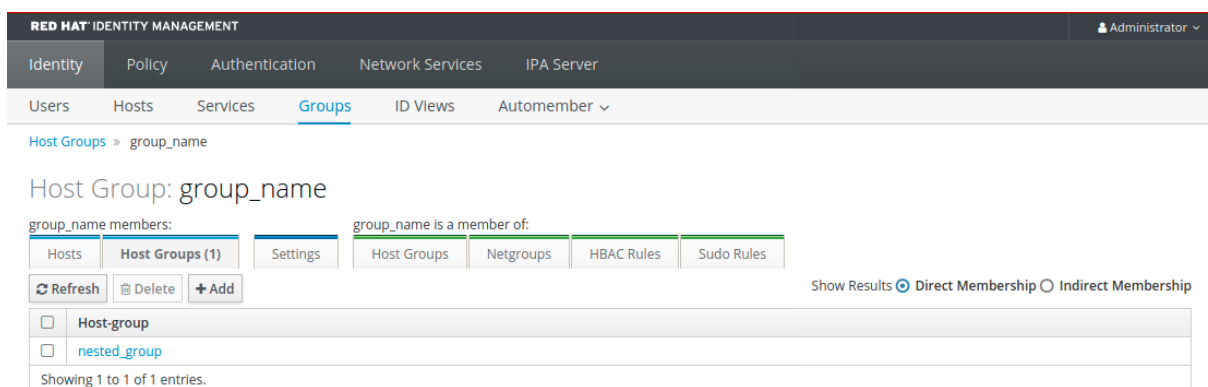
2.

목록에서 그룹을 클릭하여 이 그룹에 속한 호스트를 표시합니다. 결과를 직접 또는 간접 멤버로 제한할 수 있습니다.



3.

**Host Groups** 탭을 선택하여 이 그룹(중지 호스트 그룹)에 속하는 호스트 그룹을 표시합니다. 결과를 직접 또는 간접 멤버로 제한할 수 있습니다.



### 34.3. IDM 웹 UI에서 호스트 그룹 생성

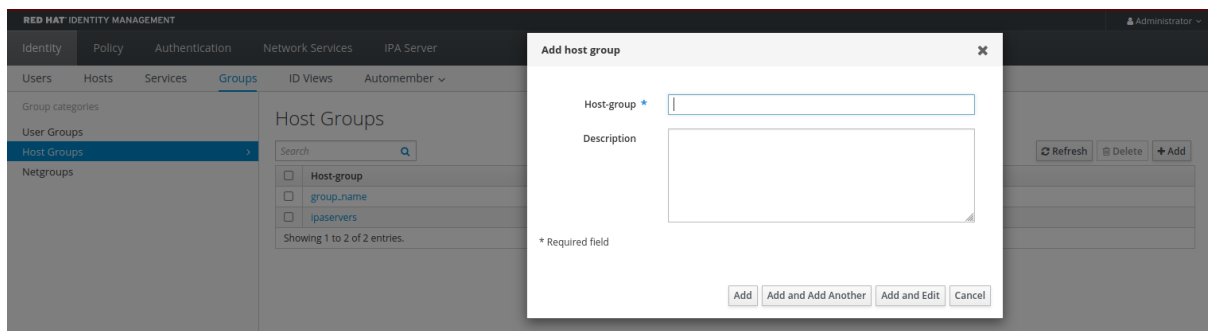
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 **IdM** 호스트 그룹을 생성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **IdM 웹 UI**에 로그인되어 있습니다. 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.

절차

1. **ID** → **그룹** 을 클릭하고 **호스트 그룹** 탭을 선택합니다.
2. **추가**를 클릭합니다. 호스트 그룹 추가 대화 상자가 나타납니다.
3. **group: name**(필수) 및 **description**(선택 사항)에 대한 정보를 입력합니다.
4. **Add** 를 클릭하여 확인합니다.



### 34.4. IDM 웹 UI에서 호스트 그룹 삭제

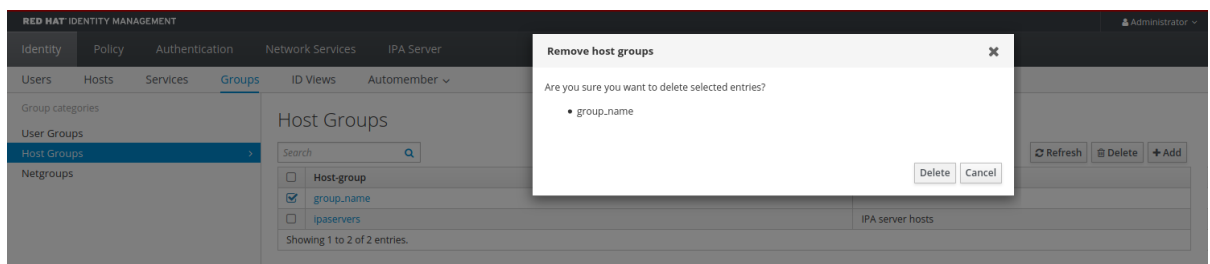
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 **IdM** 호스트 그룹을 삭제하는 방법을 설명합니다.

## 사전 요구 사항

- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **IdM 웹 UI에 로그인되어 있습니다.** 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스를 참조](#) 하십시오.

## 절차

1. **ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.**
2. **제거할 IdM 호스트 그룹을 선택하고 삭제를 클릭합니다.** 확인 대화 상자가 나타납니다.
3. **삭제를 클릭하여 확인합니다.**



## 참고

호스트 그룹을 제거해도 **IdM**에서 그룹 멤버가 삭제되지 않습니다.

## 34.5. IDM 웹 UI에 호스트 그룹 멤버 추가

이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 **IdM**에 호스트 그룹 멤버를 추가하는 방법을 설명합니다.

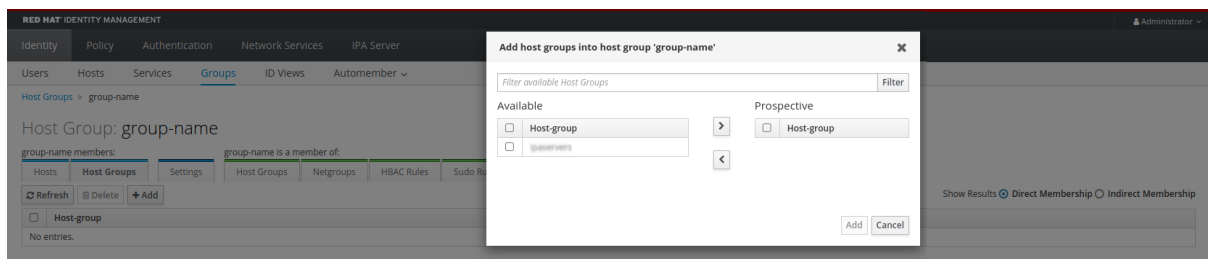
## 사전 요구 사항

- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**

- **IdM 웹 UI에 로그인되어 있습니다.** 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.

## 절차

1. **ID** → **그룹**을 클릭하고 **호스트 그룹 탭**을 선택합니다.
2. **멤버를 추가할 그룹의 이름**을 클릭합니다.
3. 추가할 **멤버 유형**에 따라 **호스트** 또는 **호스트 그룹 탭**을 클릭합니다. 해당 대화 상자가 나타납니다.
4. 추가할 **호스트** 또는 **호스트 그룹**을 선택하고 > 화살표 버튼을 클릭하여 **Prospective** 열로 이동합니다.
5. **Add**를 클릭하여 확인합니다.



## 34.6. IDM 웹 UI에서 호스트 그룹 멤버 제거

이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에서 호스트 그룹 멤버를 제거하는 방법을 설명합니다.

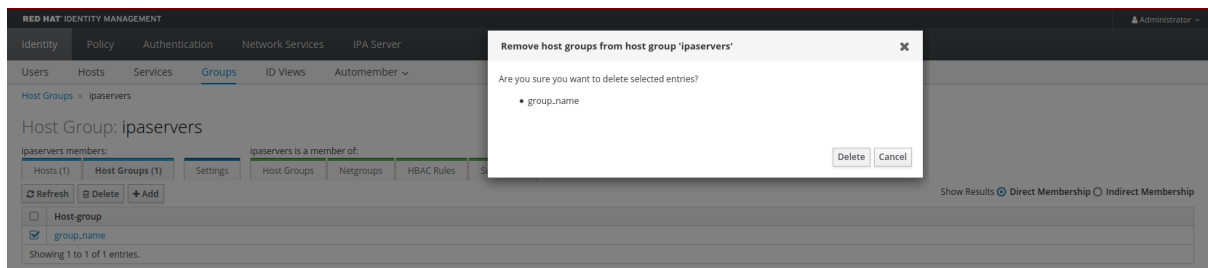
### 사전 요구 사항

- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **IdM 웹 UI에 로그인되어 있습니다.** 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스](#)를 참조

하십시오.

## 절차

1. **ID** → **그룹**을 클릭하고 **호스트 그룹** 탭을 선택합니다.
2. **멤버를 제거할 그룹의 이름**을 클릭합니다.
3. **제거할 멤버 유형**에 따라 **호스트** 또는 **호스트 그룹** 탭을 클릭합니다.
4. **제거할 멤버 옆에 있는 확인란**을 선택합니다. **Select the check box next to the member you want to remove.**
5. **삭제**를 클릭합니다. **확인 대화 상자**가 나타납니다.



6. **Delete**를 클릭하여 확인합니다. 선택한 멤버가 삭제됩니다.

## 34.7. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가

이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에서 사용자 또는 사용자 그룹을 호스트 그룹 멤버 관리자로 추가하는 방법을 설명합니다. 멤버 관리자는 호스트 그룹 멤버 관리자를 IdM 호스트 그룹에 추가할 수 있지만 호스트 그룹의 속성은 변경할 수 없습니다.

## 사전 요구 사항

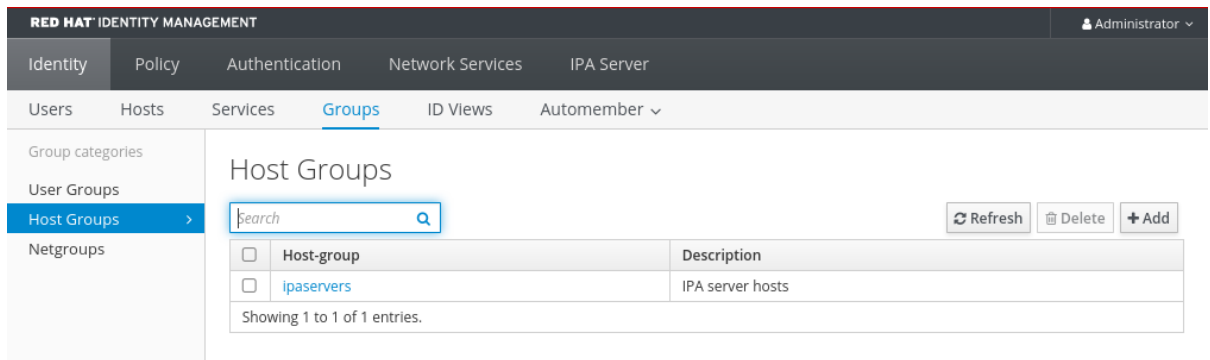
- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

- **IdM 웹 UI에 로그인되어 있습니다.** 자세한 내용은 [웹 브라우저에서 IdM 웹 UI 액세스를 참조 하십시오.](#)
- **멤버 관리자로 추가하는 호스트 그룹의 이름과 관리할 호스트 그룹의 이름이 있어야 합니다.**

## 절차

1.

**ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.**



2.

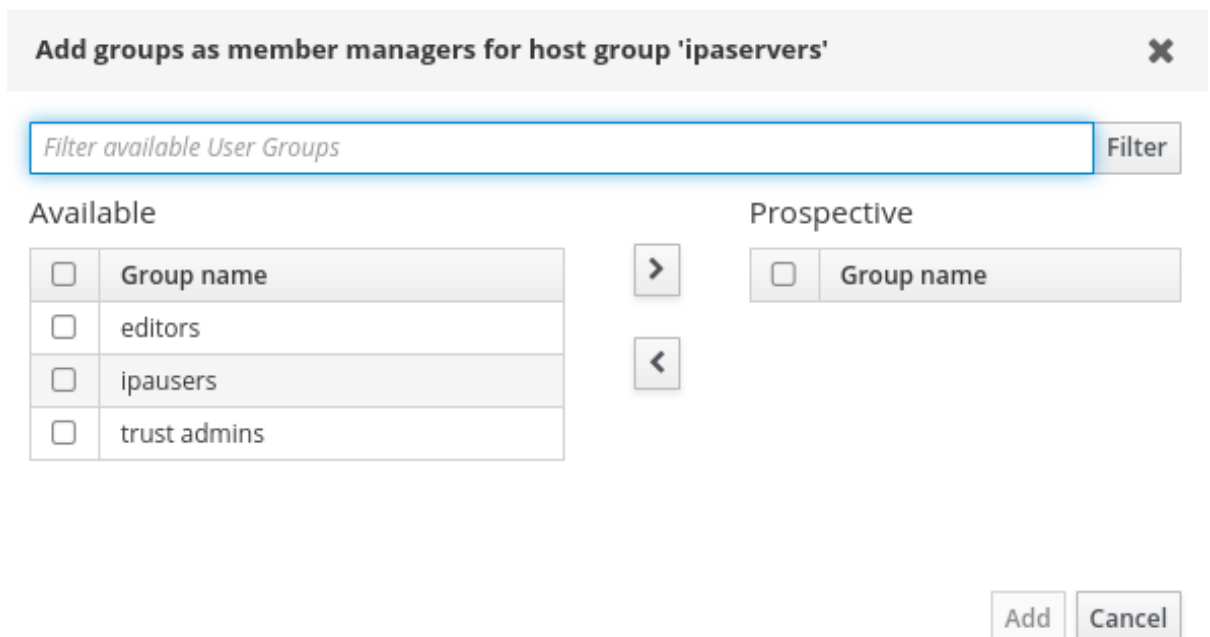
**멤버 관리자를 추가할 그룹의 이름을 클릭합니다.**

3.

**추가할 멤버 관리자 유형에 따라 멤버 관리자 탭 사용자 그룹 또는 사용자를 클릭합니다.** 해당 대화 상자가 나타납니다.

4.

**추가를 클릭합니다.**



5. 추가할 사용자 또는 사용자 그룹을 선택하고 > 화살표 버튼을 클릭하여 **Prospective** 열로 이동합니다.
6. **Add** 를 클릭하여 확인합니다.

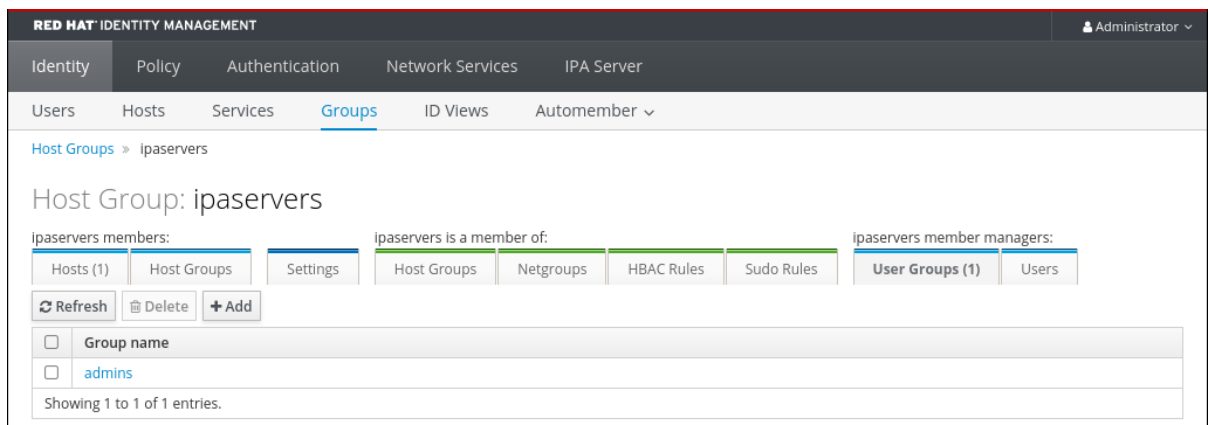


#### 참고

호스트 그룹에 멤버 관리자를 추가한 후 업데이트는 ID 관리 환경의 모든 클라이언트에 전파하는 데 시간이 걸릴 수 있습니다.

#### 검증 단계

- 호스트 그룹 대화 상자에서 사용자 그룹 또는 사용자가 그룹 또는 사용자 목록에 추가되었는지 확인합니다.



### 34.8. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거

이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에서 사용자 또는 사용자 그룹을 호스트 그룹 멤버 관리자로 제거하는 방법을 설명합니다. 멤버 관리자는 IdM 호스트 그룹에서 호스트 그룹 멤버 관리자를 제거할 수 있지만 호스트 그룹의 속성은 변경할 수 없습니다.

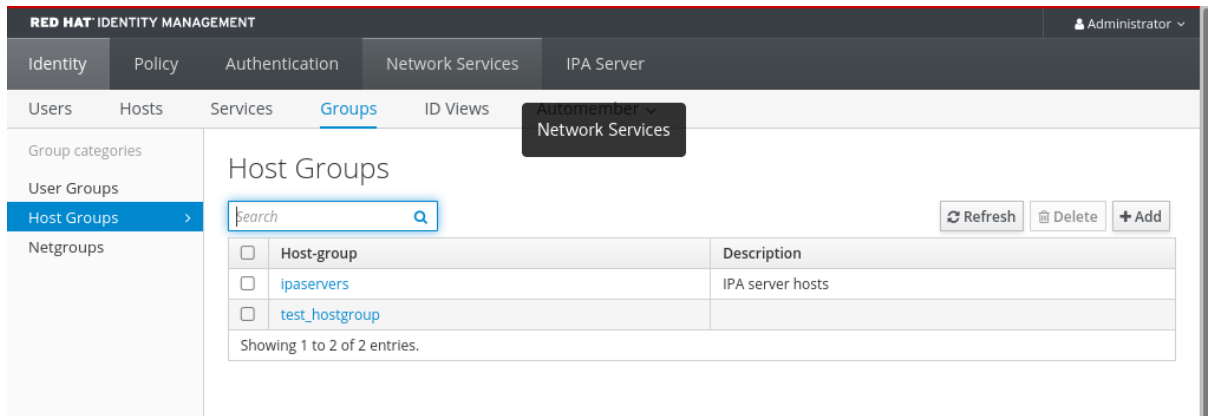
#### 사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.
-

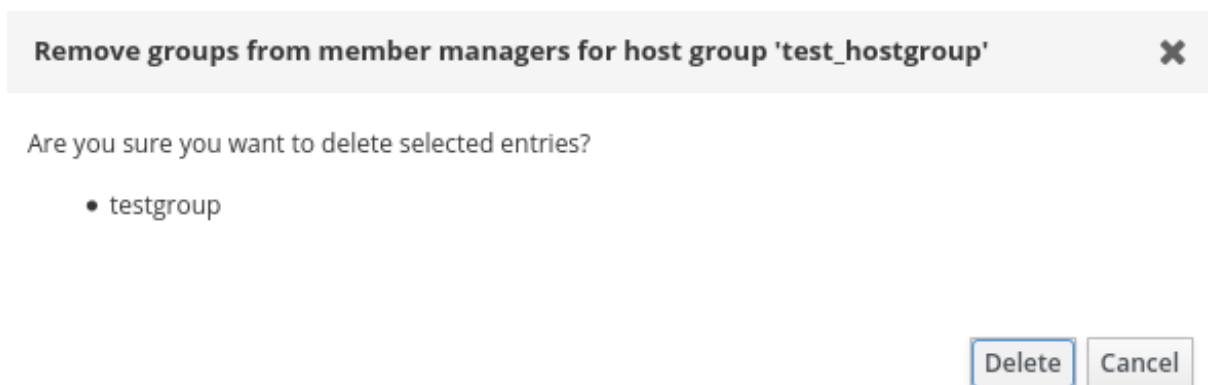
제거 중인 기존 멤버 관리자 호스트 그룹의 이름과 관리 중인 호스트 그룹의 이름이 있어야 합니다.

## 절차

1. ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.



2. 멤버 관리자를 제거할 그룹의 이름을 클릭합니다.
3. 제거할 멤버 관리자 유형에 따라 멤버 관리자 탭 사용자 그룹 또는 사용자를 클릭합니다. 해당 대화 상자가 나타납니다.
4. 제거할 사용자 또는 사용자 그룹을 선택하고 삭제를 클릭합니다.
5. **Delete** 를 클릭하여 확인합니다.



## 참고

호스트 그룹에서 멤버 관리자를 제거한 후 업데이트는 ID 관리 환경의 모든 클라이언트에 전파되는 데 시간이 걸릴 수 있습니다.

## 검증 단계

•

호스트 그룹 대화 상자에서 사용자 그룹 또는 사용자가 그룹 또는 사용자의 멤버 관리자 목록에서 제거되었는지 확인합니다.

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.

## 35장. ANSIBLE 플레이북을 사용하여 호스트 그룹 관리

이 장에서는 IdM( **I**dentit**y** **M**anagement)의 호스트 그룹을 소개하고, Ansible을 사용하여 IdM(Identity Management)의 호스트 그룹과 관련된 다음 작업을 수행하는 방법을 설명합니다.

- [IdM의 호스트 그룹](#)
- [IdM 호스트 그룹이 있는지 확인](#)
- [IdM 호스트 그룹에 호스트가 있는지 확인](#)
- [IdM 호스트 그룹 중첩](#)
- [IdM 호스트 그룹에 멤버 관리자가 있는지 확인](#)
- [IdM 호스트 그룹에서 호스트가 없는지 확인](#)
- [IdM 호스트 그룹에서 중첩 호스트 그룹이 없는지 확인](#)
- [IdM 호스트 그룹에서 멤버 관리자가 없는지 확인](#)

### 35.1. IDM의 호스트 그룹

IdM 호스트 그룹은 중요한 관리 작업, 특히 액세스 제어에 대한 제어를 중앙 집중화하는 데 사용할 수 있습니다.

#### 호스트 그룹의 정의

호스트 그룹은 공통 액세스 제어 규칙 및 기타 특성을 가진 IdM 호스트 세트가 포함된 엔티티입니다. 예를 들어 회사 부서, 물리적 위치 또는 액세스 제어 요구 사항에 따라 호스트 그룹을 정의할 수 있습니다.

IdM의 호스트 그룹에는 다음이 포함될 수 있습니다.

- **IdM 서버 및 클라이언트**
- **기타 IdM 호스트 그룹**

기본적으로 생성된 호스트 그룹

기본적으로 **IdM** 서버는 모든 **IdM** 서버 호스트에 대한 호스트 그룹 **ipaservers** 를 생성합니다.

직접 및 간접 그룹 멤버

**IdM**의 그룹 속성은 직접 및 간접 멤버 모두에 적용됩니다. 호스트 그룹 **B**가 호스트 그룹 **A**의 멤버인 경우 호스트 그룹 **B**의 모든 멤버가 호스트 그룹 **A**의 간접 멤버로 간주됩니다.

## 35.2. ANSIBLE 플레이북을 사용하여 IDENTITY 호스트 그룹이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에서 호스트 그룹이 있는지 확인하는 방법을 설명합니다.



참고

**Ansible**이 없으면 **ipa hostgroup-add** 명령을 사용하여 **IdM**에 호스트 그룹 항목이 생성됩니다. **IdM**에 호스트 그룹을 추가하는 결과는 **IdM**에 호스트 그룹 상태가 됩니다. **Ansible**의 **idempotence on idempotence** 때문에 **Ansible**을 사용하여 호스트 그룹을 **IdM**에 추가하려면 호스트 그룹의 상태를 **present: state: present** 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.

절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 타겟할 **IdM** 서버 목록으로 **ipaserver** 를 정의합니다.

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

2.

필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 예를 들어 **databases** 라는 호스트 그룹이 있는지 확인하려면 - **ipahostgroup** 작업에 **name: databases** 를 지정합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-present.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver
  become: true

  tasks:
    # Ensure host-group databases is present
    - ipahostgroup:
        ipaadmin_password: MySecret123
        name: databases
        state: present
```

플레이북에서 **state: present** 는 이미 존재하지 않는 한 호스트 그룹을 **IdM**에 추가하라는 요청을 나타냅니다.

3.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hostgroup-is-present.yml
```

## 검증 단계

1.

**admin**으로 **ipaserver** 에 로그인합니다.

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

2.

관리자를 위한 **Kerberos** 티켓을 요청합니다.

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

3.

다음은 확인하려는 IdM에 있는 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
Host-group: databases
```

IdM에 데이터베이스 호스트 그룹이 있습니다.

### 35.3. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 호스트가 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)의 호스트 그룹에 호스트가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.
- Ansible 플레이북에 참조하려는 호스트는 IdM에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 항목이 있는지 여부](#) 를 참조하십시오.
- Ansible 플레이북 파일에서 참조하는 호스트 그룹이 IdM에 추가되었습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 여부](#) 를 참조하십시오.

절차

1.

인벤토리 파일(예: `inventory.file`)을 생성하고 타겟할 IdM 서버 목록으로 `ipaserver` 를 정의합니다.

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

2.

필요한 호스트 정보로 Ansible 플레이북 파일을 생성합니다. `ipahostgroup` 변수의 `name` 매개 변수를 사용하여 호스트 그룹의 이름을 지정합니다. `ipahostgroup` 변수의 `host` 매개 변수를 사용하여 호스트 이름을 지정합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-are-hostgroups-are-in-hostgroup.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```

---
- name: Playbook to handle hostgroups
  hosts: ipaserver
  become: true

  tasks:
    # Ensure host-group databases is present
    - ipahostgroup:
        ipaadmin_password: MySecret123
        name: databases
        host:
          - db.idm.example.com
        action: member

```

이 플레이북은 **db.idm.example.com** 호스트를 **databases** 호스트 그룹에 추가합니다. **action: member** 행은 플레이북이 실행되면 **databases** 그룹 자체를 추가하려고 시도하지 않음을 나타냅니다. 대신 **db.idm.example.com** 을 데이터베이스에 추가하려는 시도만 하면 됩니다.

3. 플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-present-in-
hostgroup.yml

```

## 검증 단계

1. **admin**으로 **ipaserver** 에 로그인합니다.

```

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

```

2. 관리자를 위한 **Kerberos** 티켓을 요청합니다.

```

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

```

3. 호스트 그룹에 대한 정보를 표시하여 호스트 그룹에 있는 호스트를 확인합니다.

```

$ ipa hostgroup-show databases
Host-group: databases
Member hosts: db.idm.example.com

```

db.idm.example.com 호스트는 **databases** 호스트 그룹의 멤버로 제공됩니다.

### 35.4. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹 중첩

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 호스트 그룹에 중첩 호스트 그룹이 있는지 확인하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **Ansible** 플레이북 파일에서 참조하는 호스트 그룹은 **IdM**에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 여부](#) 를 참조하십시오.

#### 절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 타겟할 **IdM** 서버 목록으로 **ipaserver** 를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. **Ansible** 플레이북에서 호스트 그룹 **A** 가 호스트 그룹 **B**에 있는지 확인하려면 **name** 변수를 사용하여 호스트 그룹 **B**의 이름을 지정합니다. **hostgroup** 변수를 사용하여 중첩 호스트 그룹 **A**의 이름을 지정합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-are-hostgroups-are-in-hostgroup.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver
  become: true

  tasks:
    # Ensure hosts and hostgroups are present in existing databases hostgroup
    - ipahostgroup:
        ipaadmin_password: MySecret123
        name: databases
```

```

hostgroup:
- mysql-server
- oracle-server
action: member

```

이 Ansible 플레이북은 **databases** 호스트 그룹에 **mysql-server** 및 **oracle-server** 호스트 그룹이 있는지 확인합니다. **action: member** 행은 플레이북이 실행되면 IdM에 데이터베이스 그룹 자체를 추가하려고 시도하지 않음을 나타냅니다.

3. 플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-present-in-
hostgroup.yml

```

## 검증 단계

1. admin으로 ipaserver 에 로그인합니다.

```

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

```

2. 관리자를 위한 Kerberos 티켓을 요청합니다.

```

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

```

3. 중첩된 호스트 그룹이 있는 호스트 그룹에 대한 정보를 표시합니다.

```

$ ipa hostgroup-show databases
Host-group: databases
Member hosts: db.idm.example.com
Member host-groups: mysql-server, oracle-server

```

**mysql-server** 및 **oracle-server** 호스트 그룹은 **databases** 호스트 그룹에 있습니다.

## 35.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 멤버 관리자가 있는지 확인

다음 절차에서는 Ansible 플레이북을 사용하여 IdM 호스트 및 호스트 그룹에 멤버 관리자가 있는지 확

인하는 방법을 설명합니다.

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- 멤버 관리자로 추가하는 호스트 또는 호스트 그룹의 이름과 관리할 호스트 그룹의 이름이 있어야 합니다.

#### 절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 **ipaserver** 를 이 파일에 정의합니다.

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

2. 필요한 호스트 및 호스트 그룹 멤버 관리 정보로 **Ansible** 플레이북 파일을 생성합니다.

```
---
- name: Playbook to handle host group membership management
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure member manager user example_member is present for group_name
      ipahostgroup:
        ipaadmin_password: MySecret123
        name: group_name
        membermanager_user: example_member

    - name: Ensure member manager group project_admins is present for group_name
      ipahostgroup:
        ipaadmin_password: MySecret123
        name: group_name
        membermanager_group: project_admins
```

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/add-member-managers-host-groups.yml
```

## 검증 단계

**ipa group-show** 명령을 사용하여 **group\_name** 그룹에 **example\_member** 및 **project\_admins** 가 멤버 관리자로 포함되어 있는지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

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

2.

**testhostgroup** 에 대한 정보를 표시합니다.

```
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
```

## 추가 리소스

- **ipa hostgroup-add-member-manager --help** 를 참조하십시오.
- **ipa man** 페이지를 참조하십시오.

## 35.6. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 호스트가 없는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**의 호스트 그룹에서 호스트가 없는지 확인하는 방법을 설명합니다.

## 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.

- Ansible 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.
- Ansible 플레이북에 참조하려는 호스트는 IdM에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 항목이 있는지 여부](#) 를 참조하십시오.
- Ansible 플레이북 파일에서 참조하는 호스트 그룹은 IdM에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 여부](#) 를 참조하십시오.

## 절차

1.

인벤토리 파일(예: `inventory.file`)을 생성하고 타겟할 IdM 서버 목록으로 `ipaserver` 를 정의합니다.

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

2.

필요한 호스트 및 호스트 그룹 정보를 사용하여 Ansible 플레이북 파일을 생성합니다. `ipahostgroup` 변수의 `name` 매개 변수를 사용하여 호스트 그룹의 이름을 지정합니다. `ipahostgroup` 변수의 호스트 매개 변수를 사용하여 확인하려는 호스트 그룹이 없는 호스트의 이름을 지정합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-are-hostgroups-are-in-hostgroup.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver
  become: true

  tasks:
    # Ensure host-group databases is absent
    - ipahostgroup:
        ipadmin_password: MySecret123
        name: databases
        host:
          - db.idm.example.com
        action: member
        state: absent
```

이 플레이북은 `databases` 호스트 그룹에서 `db.idm.example.com` 호스트가 없는지 확인합니다. `action: member` 행은 플레이북이 실행되면 `databases` 그룹 자체를 제거하려고 시도하지 않음을 나타냅니다.

3.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-absent-in-
hostgroup.yml
```

#### 검증 단계

1.

admin으로 ipaserver 에 로그인합니다.

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

2.

관리자를 위한 Kerberos 티켓을 요청합니다.

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

3.

호스트 그룹 및 호스트에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
Host-group: databases
Member host-groups: mysql-server, oracle-server
```

db.idm.example.com 호스트는 databases 호스트 그룹에 존재하지 않습니다.

### 35.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 중복 호스트 그룹이 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)의 외부 호스트 그룹에서 중복된 호스트 그룹이 있는지 확인하는 방법을 설명합니다.

#### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.

•

**Ansible** 플레이북 파일에서 참조하는 호스트 그룹은 **IdM**에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 여부](#) 를 참조하십시오.

## 절차

1.

인벤토리 파일(예: **inventory.file**)을 생성하고 타겟할 **IdM** 서버 목록으로 **ipaserver** 를 정의합니다.

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

2.

필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. - **ipahostgroup** 변수 중에서 **name** 변수를 사용하여 외부 호스트 그룹의 이름을 지정합니다. **hostgroup** 변수를 사용하여 중첩 호스트 그룹의 이름을 지정합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-are-hostgroups-are-in-hostgroup.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver
  become: true

  tasks:
    # Ensure hosts and hostgroups are absent in existing databases hostgroup
    - ipahostgroup:
        ipaadmin_password: MySecret123
        name: databases
        hostgroup:
          - mysql-server
          - oracle-server
        action: member
        state: absent
```

이 플레이북은 **mysql-server** 및 **oracle-server** 호스트 그룹이 **databases** 호스트 그룹에 없는지 확인합니다. **action: member** 행은 플레이북이 실행되면 **IdM**에서 데이터베이스 그룹 자체가 삭제되도록 시도하지 않음을 나타냅니다.

3.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-absent-in-
hostgroup.yml
```

## 검증 단계

1. **admin**으로 **ipaserver** 에 로그인합니다.

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

2. 관리자를 위한 **Kerberos** 티켓을 요청합니다.

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

3. 중첩 호스트 그룹이 없어야 하는 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
Host-group: databases
```

출력은 **mysql-server** 및 **oracle-server** 중첩 호스트 그룹이 외부 데이터베이스 호스트 그룹에 없음을 확인합니다.

### 35.8. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹이 없는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에서 호스트 그룹이 없는지 확인하는 방법을 설명합니다.



#### 참고

**Ansible**이 없으면 **ipa hostgroup-del** 명령을 사용하여 **IdM**에서 호스트 그룹 항목이 제거됩니다. **IdM**에서 호스트 그룹을 제거하는 결과는 **IdM**에 호스트 그룹이 없는 상태입니다. **Ansible**의 **idempotence on idempotence** 때문에 **Ansible**을 사용하여 **IdM**에서 호스트 그룹을 제거하려면 호스트 그룹의 상태를 **absent: state: absent** 로 정의하는 플레이북을 생성해야 합니다.

#### 사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.

## 절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 타겟할 IdM 서버 목록으로 `ipaserver` 를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-absent.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver
  become: true

  tasks:
  - Ensure host-group databases is absent
    ipahostgroup:
      ipaadmin_password: MySecret123
      name: databases
      state: absent
```

이 플레이북은 IdM에서 **databases** 호스트 그룹이 없는지 확인합니다. **state: absent** 는 이미 삭제되지 않는 한 IdM에서 호스트 그룹을 삭제하라는 요청을 의미합니다.

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hostgroup-is-absent.yml
```

## 검증 단계

1. **admin**으로 `ipaserver` 에 로그인합니다.

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

2. 관리자를 위한 **Kerberos** 티켓을 요청합니다.

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

3. 확인하지 않은 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
ipa: ERROR: databases: host group not found
```

IdM에는 데이터베이스 호스트 그룹이 없습니다.

### 35.9. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 멤버 관리자가 없는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 호스트 및 호스트 그룹에 멤버 관리자가 없는지 확인하는 방법을 설명합니다.

#### 사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.
- 멤버 관리자로 제거하고 관리하는 호스트 그룹의 이름을 제거하는 사용자 또는 사용자 그룹의 이름이 있어야 합니다.

#### 절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 `ipaserver` 를 이 파일에 정의합니다.

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

2. 필요한 호스트 및 호스트 그룹 멤버 관리 정보로 **Ansible** 플레이북 파일을 생성합니다.

```
---
- name: Playbook to handle host group membership management
  hosts: ipaserver
  become: true
```

```

tasks:
- name: Ensure member manager host and host group members are absent for
  group_name
  ipahostgroup:
    ipaadmin_password: MySecret123
    name: group_name
    membermanager_user: example_member
    membermanager_group: project_admins
    action: member
    state: absent

```

3.

플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-member-managers-host-groups-are-absent.yml

```

### 검증 단계

**ipa group-show** 명령을 사용하여 **group\_name** 그룹에 **example\_member** 또는 **project\_admins** 가 멤버 관리자로 포함되어 있지 않은지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

```

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

```

2.

**testhostgroup** 에 대한 정보를 표시합니다.

```

ipaserver]$ ipa hostgroup-show group_name
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: testhostgroup2

```

### 추가 리소스

- **ipa hostgroup-add-member-manager --help** 를 참조하십시오.
- **ipa man** 페이지를 참조하십시오.

## 36장. ANSIBLE 플레이북을 사용하여 IDM에 호스트 기반 액세스 제어 규칙이 있는지 확인

이 장에서는 IdM(Identity Management) 호스트 기반 액세스 정책과 **Ansible** 을 사용하여 정의하는 방법을 설명합니다.

**Ansible**은 시스템 구성, 소프트웨어 배포 및 롤링 업데이트를 수행하는 데 사용되는 자동화 틀입니다. IdM(Identity Management) 지원이 포함됩니다.

### 36.1. IDM의 호스트 기반 액세스 제어 규칙

호스트 기반 액세스 제어(HBAC) 규칙은 서비스 그룹의 서비스 또는 서비스를 사용하여 호스트 또는 호스트 그룹에 액세스할 수 있는 사용자 또는 사용자 그룹을 정의합니다. 시스템 관리자는 HBAC 규칙을 사용하여 다음 목표를 달성할 수 있습니다.

- 도메인의 지정된 시스템에 대한 액세스 권한을 특정 사용자 그룹의 멤버로 제한합니다.
- 특정 서비스만 도메인의 시스템에 액세스하는 데 사용할 수 있도록 허용합니다.

기본적으로 IdM은 전체 IdM 도메인의 모든 관련 서비스를 통해 모든 사용자에게 모든 호스트에 대한 범용 액세스를 의미하는 **allow\_all** 이라는 기본 HBAC 규칙으로 구성됩니다.

기본 **allow\_all** 규칙을 고유한 HBAC 규칙 세트로 대체하여 다른 호스트에 대한 액세스를 미세 조정할 수 있습니다. 중앙 집중식으로 간소화된 액세스 제어 관리의 경우 HBAC 규칙을 개별 사용자, 호스트 또는 서비스 대신 사용자 그룹, 호스트 그룹 또는 서비스 그룹에 적용할 수 있습니다.

### 36.2. ANSIBLE 플레이북을 사용하여 IDM에 HBAC 규칙이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 IdM(Identity Management)에 호스트 기반 액세스 제어(HBAC) 규칙이 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **ansible-freeipa** 패키지는 **Ansible** 컨트롤러에 설치됩니다.

- **IdM 관리자 암호를 알고 있습니다.**
- **HBAC 규칙에 사용할 사용자 및 사용자 그룹은 IdM에 있습니다.** 자세한 내용은 [Ansible 플레이북을 사용하여 사용자 계정 관리](#) 및 [Ansible 플레이북을 사용하여 IdM 그룹 및 그룹 멤버가 있는지 확인](#)합니다.
- **HBAC 규칙을 적용할 호스트 및 호스트 그룹이 IdM에 있습니다.** 자세한 내용은 [Ansible 플레이북을 사용하여 호스트 관리](#) 및 [Ansible 플레이북을 사용하여 호스트 그룹을 관리](#)하십시오.

## 절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 `ipaserver` 를 이 파일에 정의합니다.

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

2. 확인하려는 **HBAC 정책**을 정의하는 **Ansible 플레이북** 파일을 생성합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/hbacrule/ensure-hbacrule-allhosts-present.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Playbook to handle hbacrules
  hosts: ipaserver
  become: true

  tasks:
    # Ensure idm_user can access client.idm.example.com via the sshd service
    - ipahbacrule:
        ipaadmin_password: MySecret123
        name: login
        user: idm_user
        host: client.idm.example.com
        hbacsvc:
          - sshd
        state: present
```

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-new-hbacrule-present.yml
```

## 검증 단계

1. 관리자 권한으로 **IdM 웹 UI**에 로그인합니다.
2. 정책 → 호스트 기반 액세스 제어 → **HBAC** 테스트 로 이동합니다.
3. **who** 탭에서 **idm\_user**를 선택합니다.
4. 액세스 탭에서 **client.idm.example.com** 을 선택합니다.
5. **Via** 서비스 탭에서 **sshd** 를 선택합니다.
6. 규칙 탭에서 로그인 을 선택합니다.
7. **Run test** (테스트 실행) 탭에서 **Run test**( 테스트 실행 ) 버튼을 클릭합니다. **ACCESS GRANTED**가 표시되면 **HBAC** 규칙이 성공적으로 구현됩니다.

#### 추가 리소스

- **/usr/share/doc/ansible-freeipa** 디렉터리의 **README-hbacsvgroup.md**, **README-hbacrule.md** 파일을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/playbooks** 디렉터리의 하위 디렉터리에 있는 플레이북을 참조하십시오.

## 37장. 짧은 AD 사용자 이름을 확인하도록 도메인 확인 순서 구성

기본적으로 AD(Active Directory) 환경에서 사용자와 그룹을 확인하고 인증하려면 `user_name@domain.com` 또는 `domain.com\user_name` 형식으로 정규화된 이름을 지정해야 합니다. 다음 섹션에서는 짧은 AD 사용자 이름과 그룹 이름을 확인하도록 IdM 서버 및 클라이언트를 구성하는 방법을 설명합니다.

- 도메인 확인 순서의 작동 방식
- IdM 서버에서 글로벌 도메인 확인 순서 설정
- IdM 서버에서 ID 보기의 도메인 확인 순서 설정
- IdM 클라이언트의 SSSD에서 도메인 확인 순서 설정

### 37.1. 도메인 확인 순서의 작동 방식

AD(Active Directory)가 신뢰할 수 있는 IdM(Identity Management) 환경에서는 정규화된 이름을 지정하여 사용자와 그룹을 확인하고 인증할 것을 권장합니다. 예를 들어 다음과 같습니다.

- `<idm_username>` `idm.example.com` 의 IdM 사용자의 경우 `idm.example.com`
- `<ad_username>` `ad.example.com` 도메인에서 AD 사용자의 경우 `@ ad.example.com`

기본적으로 `ad_username` 과 같은 짧은 이름 형식을 사용하여 사용자 또는 그룹 조회를 수행하는 경우 IdM은 IdM 도메인만 검색하고 AD 사용자 또는 그룹을 찾지 못합니다. 짧은 이름을 사용하여 AD 사용자 또는 그룹을 확인하려면 도메인 확인 순서 옵션을 설정하여 IdM이 여러 도메인을 검색하는 순서를 변경합니다.

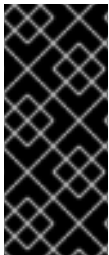
IdM 데이터베이스에서 도메인 확인 순서를 중앙에서 설정하거나 개별 클라이언트의 SSSD 구성에서 설정할 수 있습니다. IdM은 다음과 같은 우선순위 순서로 도메인 확인 순서를 평가합니다.

- 로컬 `/etc/sss/sss.conf` 설정.

- ID 보기 구성입니다.
- 글로벌 IdM 구성.

## 참고

- 호스트의 SSSD 구성에 **default\_domain\_suffix** 옵션이 포함되어 있고 이 옵션으로 지정되지 않은 도메인에 대한 요청을 만들려면 정규화된 사용자 이름을 사용해야 합니다.
- 도메인 확인 순서 옵션을 사용하고 **compat** 트리를 쿼리하는 경우 여러 개의 UID(사용자 ID)가 수신될 수 있습니다. 이 경우 **도메인 확인 순서가 설정된 경우 [Pagure 버그 report](#) Inconsistent compat 사용자 개체를 AD 사용자**의 사용자 개체를 참조하십시오.



## 중요

IdM 클라이언트 또는 IdM 서버에 **full\_name\_format SSSD** 옵션을 사용하지 마십시오. 이 옵션에 기본값이 아닌 값을 사용하면 사용자 이름이 표시되는 방식을 변경하고 IdM 환경에서 조화가 중단될 수 있습니다.

## 추가 리소스

- 레거시 [Linux](#) 클라이언트를 위한 [Active Directory](#) 트러스트.

## 37.2. IDM 서버에서 글로벌 도메인 확인 순서 설정

이 절차에서는 IdM 도메인에 있는 모든 클라이언트의 도메인 확인 순서를 설정합니다. 이 예에서는 다음 순서로 사용자와 그룹을 검색하도록 도메인 확인 순서를 설정합니다.

1. **Active Directory (AD) 루트 도메인 `ad.example.com`**
2. **AD 하위 도메인 `subdomain1.ad.example.com`**
3. **IdM 도메인 `idm.example.com`**

## 사전 요구 사항

- **AD 환경에 대한 신뢰를 구성했습니다.**

## 절차

- **ipa config-mod --domain-resolution-order** 명령을 사용하여 기본 순서로 검색할 도메인을 나열합니다. 도메인을 콜론(:)으로 구분합니다.

```
[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
...
```

## 검증 단계

- 짧은 이름 만 사용하여 **ad.example.com** 도메인에서 사용자에게 대한 사용자 정보를 검색할 수 있는지 확인합니다.

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

### 37.3. IDM 서버에서 ID 보기의 도메인 확인 순서 설정

이 절차에서는 특정 **IdM** 서버 및 클라이언트 세트에 적용할 수 있는 **ID** 보기의 도메인 확인 순서를 설정합니다. 이 예제에서는 **IdM** 호스트 **client1.idm.example.com** 에 대해 **ADsubdomain1\_first** 이라는 **ID** 보기를 생성하고 다음 순서로 사용자와 그룹을 검색하기 위해 도메인 확인을 설정합니다.

1. **Active Directory (AD)** 하위 도메인 **subdomain1.ad.example.com**
2. **AD root** 도메인 **ad.example.com**
3. **IdM** 도메인 **idm.example.com**



## 참고

ID 보기에 설정된 도메인 확인 순서는 전역 도메인 확인 순서를 재정의하지만 **SSSD** 구성에서 로컬로 설정된 도메인 확인 순서를 재정의하지는 않습니다.

## 사전 요구 사항

- **AD** 환경에 대한 신뢰를 구성했습니다.

## 절차

1. **--domain-resolution-order** 옵션을 설정하여 ID 보기를 생성합니다.

```
[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. **IdM** 호스트에 ID 보기를 적용합니다.

```
[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
-----
```

## 검증 단계

- ID 보기의 세부 정보를 표시합니다.

```
[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**

- 짧은 이름 만 사용하여 **subdomain1.ad.example.com** 도메인에서 사용자에게 대한 사용자 정보를 검색할 수 있는지 확인합니다.

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

### 37.4. IDM 클라이언트의 SSSD에서 도메인 확인 순서 설정

이 절차에서는 **IdM** 클라이언트의 **SSSD** 구성에 도메인 확인 순서를 설정합니다. 이 예제에서는 **IdM** 호스트 **client2.idm.example.com** 을 구성하여 다음 순서로 사용자와 그룹을 검색합니다.

1. **Active Directory (AD)** 하위 도메인 **subdomain1.ad.example.com**
2. **AD root** 도메인 **ad.example.com**
3. **IdM** 도메인 **idm.example.com**



참고

로컬 **SSSD** 구성의 도메인 확인 순서는 글로벌 및 **ID** 보기 도메인 확인 순서를 재정의합니다.

사전 요구 사항

- **AD** 환경에 대한 신뢰를 구성했습니다.

절차

1. 텍스트 편집기에서 **/etc/sss/sss.conf** 파일을 엽니다.
2. 파일의 **[sss]** 섹션에서 **domain\_resolution\_order** 옵션을 설정합니다.

```
domain_resolution_order = subdomain1.ad.example.com, ad.example.com,
idm.example.com
```

3. 파일을 저장한 후 닫습니다.
4. **SSSD** 서비스를 다시 시작하여 새 구성 설정을 로드합니다.

```
[root@client2 ~]# systemctl restart sssd
```

#### 검증 단계

- 짧은 이름 만 사용하여 **subdomain1.ad.example.com** 도메인에서 사용자에게 대한 사용자 정보를 검색할 수 있는지 확인합니다.

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

### 37.5. 추가 리소스

- [ID 보기를 사용하여 IdM 클라이언트의 사용자 속성 값 덮어쓰기](#)

## 38장. IDM에서 AD 사용자 계정 이름을 사용하여 인증 활성화

### 38.1. IDM에서 신뢰할 수 있는 AD FOREST의 사용자 보안 주체 이름

IdM(Identity Management) 관리자는 AD 사용자가 대체 UPN( User Principal Names )을 사용하여 IdM 도메인의 리소스에 액세스하도록 허용할 수 있습니다. UPN은 AD 사용자가 `user_name@KERBEROS-REALM` 형식으로 인증하는 대체 사용자 로그인입니다. AD 관리자는 AD forest에서 추가 Kerberos 별칭과 UPN 접미사를 모두 구성할 수 있으므로 `user_name` 및 `KERBEROS-REALM` 둘 다에 대한 대체 값을 설정할 수 있습니다.

예를 들어 회사에서 Kerberos 영역 `AD.EXAMPLE.COM` 을 사용하는 경우 사용자의 기본 UPN은 `user@ad.example.com` 입니다. 사용자가 이메일 주소(예: `user@example.com`)를 사용하여 로그인할 수 있도록 하려면 `EXAMPLE.COM` 을 AD에서 대체 UPN으로 구성할 수 있습니다. 회사에서 최근 병합을 경험하고 사용자에게 통합 로그인 네임스페이스를 제공하려는 경우 대체 *UPN(Enterprise UPNs*라고도 함)이 특히 편리합니다.

UPN 접미사는 AD forest root에 정의된 경우에만 IdM에 대해 표시됩니다. AD 관리자는 Active Directory 도메인 및 신뢰 유틸리티 또는 PowerShell 명령줄 도구를 사용하여 UPN을 정의할 수 있습니다.

#### 참고

사용자에 대해 UPN 접미사를 구성하려면 Active Directory 도메인 및 신뢰 유틸리티와 같은 오류 검증을 수행하는 툴을 사용하는 것이 좋습니다.

Active Directory가 해당 작업을 확인하지 않기 때문에 `ldapmodify` 명령을 사용하여 사용자에게 대해 `userPrincipalName` 특성을 설정하는 등의 낮은 수준의 수정을 통해 UPN을 구성하는 것이 좋습니다.

AD 측에 새 UPN을 정의한 후 IdM 서버에서 `ipa trust-fetch-domains` 명령을 실행하여 업데이트된 UPN을 검색합니다. **IdM에서 AD UPN이 최신 상태를** 확인합니다.

IdM은 도메인에 대한 UPN 접미사를 도메인의 하위 트리 `cn=trusted_domain_name,cn=ad,cn=trusts,dc=idm,dc=example,dc=com` 에 저장합니다.

#### 추가 리소스



**AD forest 루트에서 UPN 접미사 설정을 스크립팅하는 방법**

- **AD 사용자 항목을 수동으로 수정하고 UPN 접미사 검증을 우회하는 방법**
- **신뢰 컨트롤러 및 신뢰 에이전트**

### 38.2. IDM에서 AD UPNS가 최신 버전인지 확인

신뢰할 수 있는 **AD(Active Directory) forest**에서 **UPN(User Principal Name)** 접미사를 추가하거나 제거한 후 **IdM** 서버에서 신뢰할 수 있는 포리스트에 대한 정보를 새로 고칩니다.

#### 사전 요구 사항

- **IdM 관리자 자격 증명.**

#### 절차

- **ipa trust-fetch-domains** 명령을 입력합니다. 예상 빈 출력은 다음과 같습니다.

```
[root@ipaserver ~]# ipa trust-fetch-domains
Realm-Name: ad.example.com
-----
No new trust domains were found
-----
Number of entries returned 0
-----
```

#### 검증 단계

- **ipa trust-show** 명령을 입력하여 서버가 새 **UPN**을 가져온지 확인합니다. 메시지가 표시되면 **AD** 영역의 이름을 지정합니다.

```
[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
```

출력에서 **example.com UPN** 접미사가 이제 **ad.example.com** 영역 항목의 일부임을 보여줍니다.

### 38.3. AD UPN 인증 문제에 대한 문제 해결 데이터 수집

이 절차에서는 **AD(Active Directory)** 환경 및 **IdM** 환경에서 **UPN(User Principal Name)** 구성에 대한 문제 해결 데이터를 수집하는 방법을 설명합니다. **AD** 사용자가 대체 **UPN**을 사용하여 로그인할 수 없는 경우 이 정보를 사용하여 문제 해결 작업을 줄일 수 있습니다.

#### 사전 요구 사항

- **AD** 도메인 컨트롤러에서 정보를 검색하려면 **IdM** 신뢰 컨트롤러 또는 신뢰 에이전트에 로그인해야 합니다.
- 다음 구성 파일을 수정하고 **IdM** 서비스를 다시 시작하려면 **root** 권한이 필요합니다.

#### 절차

1. 텍스트 편집기에서 **/usr/share/ipa/smb.conf.empty** 구성 파일을 엽니다.

2. 파일에 다음 콘텐츠를 추가합니다.

```
[global]
log level = 10
```

3. **/usr/share/ipa/smb.conf.empty** 파일을 저장하고 종료합니다.

4. 텍스트 편집기에서 **/etc/ipa/server.conf** 구성 파일을 엽니다. 해당 파일이 없는 경우 새로 만듭니다.

5. 파일에 다음 콘텐츠를 추가합니다.

```
[global]
debug = True
```

6. **/etc/ipa/server.conf** 파일을 저장하고 닫습니다.

7.

**Apache webserver** 서비스를 재시작하여 구성 변경 사항을 적용합니다.

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

8.

**AD** 도메인에서 신뢰 정보를 검색합니다.

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

9.

다음 로그 파일에서 디버깅 출력 및 문제 해결 정보를 검토합니다.

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

#### 추가 리소스

- [AD UPN 인증 문제에 대한 문제 해결 데이터를 수집하려면 Using-02-client](#) 를 참조하십시오.

## 39장. AD 사용자가 IDM을 관리할 수 있도록 활성화

### 39.1. AD 사용자의 ID 덮어쓰기

**AD(Active Directory)** 사용자 및 그룹 액세스를 **IdM** 그룹의 멤버로 **AD** 사용자에게 대한 **ID** 사용자 재정의의 추가하여 **POSIX** 환경에서 **IdM(Identity Management)** 리소스에 대한 액세스를 중앙에서 관리할 수 있습니다.

**ID** 재정의는 특정 **ID** 보기(이 경우 기본 신뢰 보기 ) 내에서 특정 **Active Directory** 사용자 또는 그룹 속성이 표시되는 것을 설명하는 레코드입니다. 이 기능을 통해 **IdM LDAP** 서버는 **IdM** 그룹에 대한 액세스 제어 규칙을 **AD** 사용자에게 적용할 수 있습니다.

예를 들어 **AD** 사용자는 **IdM UI**의 셀프 서비스 기능을 사용하여 **SSH** 키를 업로드하거나 개인 데이터를 변경할 수 있습니다. **AD** 관리자가 두 개의 서로 다른 계정 및 암호 없이도 **IdM**을 완전히 관리할 수 있습니다.



참고

현재는 **AD** 사용자가 **IdM**에서 선택된 기능을 사용할 수 없습니다. 예를 들어 **IdM** 관리자 그룹의 **AD** 사용자로 **IdM** 사용자의 암호를 설정하는 데 실패할 수 있습니다.

추가 리소스

- [Active Directory 사용자를 위한 ID 보기 사용](#)

### 39.2. ID 덮어쓰기를 사용하여 AD 사용자가 IDM 관리 가능

이 절차에서는 **AD** 사용자가 **ID** 재정의의 생성하고 사용하여 **IdM** 사용자의 사용자와 동일한 사용자 권한을 부여하는 방법을 설명합니다. 이 절차 중에 신뢰 컨트롤러 또는 신뢰 에이전트로 구성된 **IdM** 서버에서 작업합니다.

사전 요구 사항

- 작동하는 **IdM** 환경이 설정되어 있습니다. 자세한 내용은 [Identity Management 설치](#)를 참조하십시오.
- **IdM** 환경과 **AD** 간의 작동 신뢰가 설정됩니다.

## 절차

1.

IdM 관리자로서 기본 신뢰 보기의 AD 사용자에게 대한 ID를 생성합니다. 예를 들어 `ad_user@ad.example.com` 사용자에게 대한 ID 덮어쓰기를 생성하려면 다음을 수행합니다.

```
# kinit admin
# ipa idoverrideuser-add 'default trust view' ad_user@ad.example.com
```

2.

기본 신뢰 보기의 ID 재정의의 IdM 그룹에 멤버로 추가합니다. 질문의 그룹이 IdM 역할의 멤버인 경우 ID 재정의로 표시되는 AD 사용자는 명령줄 인터페이스와 IdM 웹 UI를 포함하여 IdM API를 사용할 때 역할에 의해 부여된 모든 권한을 얻습니다. 예를 들어 `ad_user@ad.example.com` 사용자의 ID 덮어쓰기를 `admins` 그룹에 추가하려면 다음을 수행합니다.

```
# ipa group-add-member admins --idoverrideusers=ad_user@ad.example.com
```

## 추가 리소스

•

[Active Directory](#) 사용자를 위한 ID 보기 사용

## 39.3. AD 사용자로 IDM CLI 관리

이 절차에서는 AD(Active Directory) 사용자가 IdM(Identity Management) 명령줄 인터페이스(CLI)에 로그인할 수 있는지 확인하고 해당 역할에 적합한 명령을 실행합니다.

1.

IdM 관리자의 현재 Kerberos 티켓을 삭제합니다.

```
# kdestroy -A
```



## 참고

MIT Kerberos의 GSSAPI 구현에서 기본 설정별로 대상 서비스의 영역에서 자격 증명을 선택하기 때문에 Kerberos 티켓을 제거해야 합니다. 이 경우 IdM 영역입니다. 즉, 인증 정보 캐시 컬렉션, 즉 `KCM:`, `KEYRING:`, `DIR:` 자격 증명 캐시 유형이 사용 중인 경우 이전에 가져온 `admin` 또는 기타 IdM 주체의 인증 정보가 AD 사용자 자격 증명 대신 IdM API에 액세스하는 데 사용됩니다.

2.

ID 덮어쓰기를 만든 AD 사용자의 Kerberos 자격 증명을 가져옵니다.

```
# kinit ad_user@AD.EXAMPLE.COM
Password for ad_user@AD.EXAMPLE.COM:
```

3.

**AD** 사용자의 **ID** 재정의가 해당 그룹의 **IdM** 사용자와 동일한 권한을 갖는지 테스트합니다.  
**AD** 사용자의 **ID** 재정의가 **admins** 그룹에 추가된 경우, 예를 들어 **AD** 사용자는 **IdM**에 그룹을 생성할 수 있습니다.

```
# ipa group-add some-new-group
-----
Added group "some-new-group"
-----
Group name: some-new-group
GID: 1997000011
```